Fundamentals of Electrical Engineering

Don H. Johnson
J.D. Wise, Jr.

Rice University

© 1999

Contents

1 **Introduction** **1**
- 1.1 Themes 1
- 1.2 What is Information? 3
  - 1.2.1 Analog Information 3
  - 1.2.2 Digital Information 4
- 1.3 Communication Systems 4
- 1.4 The Fundamental Signal 7
- 1.5 Communicating Information with Signals 8
- Problems 9

2 **Signals and Systems** **11**
- 2.1 Signal Theory 11
- 2.2 System Theory 15
  - 2.2.1 Connecting Systems Together 15
  - 2.2.2 Simple Systems 16
- 2.3 Signal Decomposition 19
- Problems 20

3 **Analog Signal Processing** **23**
- 3.1 Voltage and Current 23
- 3.2 Linear Circuit Elements 24
- 3.3 Sources 25
- 3.4 Ideal and Real-World Circuit Elements 25
- 3.5 Electric Circuits and Interconnection Laws 26
- 3.6 Equivalent Circuits 35
- 3.7 Circuits with Capacitors and Inductors 39
  - 3.7.1 Impedance 39
  - 3.7.2 Transfer Function 43

3.7.3 Equivalent Circuits Revisited 45
3.7.4 Summary 47
3.8 Formal Circuit Methods 49
3.9 Circuit Models for Communication Channels 53
3.9.1 Wireline Channels 55
3.9.2 Wireless Channels 60
3.10 Electronics 62
3.10.1 Dependent Sources 62
3.10.2 Operational Amplifiers 65
3.11 Nonlinear Circuit Elements 70
Problems 72

4 The Frequency Domain 83
4.1 Fourier Series 83
4.1.1 A Signal's Spectrum 86
4.1.2 Equality of Representations 90
4.2 Complex Fourier Series 94
4.3 Encoding Information in the Frequency Domain 98
4.4 Filtering Periodic Signals 100
4.5 The Fourier Transform 102
4.6 Linear, Time-Invariant Systems 109
4.7 Modeling the Speech Signal 112
4.8 The Sampling Theorem 119
4.9 Analog-to-Digital Conversion 122
Problems 123

5 Digital Signal Processing 131
5.1 Discrete-Time Signals and Systems 131
5.1.1 Real- and complex-valued signals 132
5.1.2 Symbolic Signals 133
5.2 Digital Processing Systems 134
5.3 Discrete-Time Fourier Transform (DTFT) 137
5.4 Discrete Fourier Transform (DFT) 144
5.4.1 Computational Complexity 146
5.4.2 The Fast Fourier Transform 147
5.4.3 Spectrograms 150
5.5 Discrete-Time Systems 153
5.5.1 Systems in the Time-Domain 154
5.5.2 Systems in the Frequency Domain 158

5.6 Discrete-Time Filtering of Analog Signals 160
5.7 Modeling Real-World Systems with Artificial Ones 170
Problems 172

6 Information Communication 179
6.1 Communication Channels 180
6.1.1 Noise and Interference 183
6.1.2 Channel Models 184
6.2 Analog Communication 185
6.2.1 Baseband communication 186
6.2.2 Modulated Communication 186
6.3 Digital Communication 190
6.3.1 Signal Sets 191
6.3.2 Digital Communication Receivers 195
6.3.3 Digital Channels 201
6.3.4 Digital Sources and Compression 202
6.3.5 Error-Correcting Codes: Channel Coding 208
6.4 Comparison of Analog and Digital Communication 219
6.5 Communication Networks 220
6.5.1 Message routing 223
6.5.2 Network architectures and interconnection 224
6.5.3 Communication protocols 228
Problems 230

A Complex Numbers 239
A.1 Definitions 239
A.2 Arithmetic in Cartesian Form 241
A.3 Arithmetic in Polar Form 241
Problems 242

B Decibels 243
Problems 244

C Frequency Allocations 245

Answers to Questions 247

Chapter 1

Introduction

1.1 Themes

From its beginnings in the late nineteenth century, electrical engineering has blossomed from its focus on electrical circuits for power, telegraphy and telephony to a much broader range of disciplines. Still, the underlying themes are relevant today: *Power* creation and transmission and *information* have run through electrical engineering's century and a half life. This book concentrates on the latter theme: the *representation, manipulation, transmission, and reception of information by electrical means.* This book describes what information is, how engineers quantify information, and how electrical *signals* represent information.

Information has a variety of forms. When you speak to a friend, your thoughts are translated by your brain into motor commands that cause various vocal tract components—the jaw, the tongue, the lips —to move in a coordinated fashion. Information arises in your thoughts and is represented by speech, which must have a well defined, broadly known structure so that someone else can understand what you say. Utterances convey information in sound pressure waves, which propagate to your friend's ear. There, sound energy is converted back to neural activity, and, if what you say makes sense, she understands what you say. Your words could have been recorded on a compact disc (CD), mailed to your friend and listened by her on her stereo. Information can take the form of a text file you type into your word processor. You might send the file via e-mail to a friend, who reads it and understands it. From an information theoretic viewpoint, all of these scenarios are equivalent, although the forms of the information representation —sound waves, plastic and computer files —are very different.

The conversion of information-bearing signals from one energy form into another is known as *energy conversion* or *transduction.* All conversion systems are inefficient it that some input energy is lost as heat, but this loss does not necessarily mean that the information conveyed is lost. Conceptually we could use any form of energy to represent information, but electric signals are uniquely well-suited for information representation, transmission (signals can be broadcast from antennas or sent through wires), and manipulation (circuits can be built to reduce noise and computers can be used to modify information). Thus, we will be concerned with how to

- *represent* all forms of information with electrical signals,
- *encode* information as voltages, currents, and electromagnetic waves,
- *manipulate* information-bearing electric signals with circuits and computers, and
- *receive* electric signals and convert the information expressed by electric signals back into a useful form.

Telegraphy represents the earliest electrical information system, and it dates from 1837. At that time, electrical science was largely empirical, and only those with experience and intuition could develop telegraph systems. Electrical science came of age when James Clerk Maxwell proclaimed in 1864 a set of equations that he claimed governed all electrical phenomena. These equations predicted that light was an electromagnetic wave, and that energy could propagate. Because of the complexity of Maxwell's presentation, largely empirical work forged the development of the telephone in 1876. Once Heinrich Hertz confirmed Maxwell's prediction of what we now call radio waves in about 1882, Maxwell's equations were simplified by Oliver Heaviside and others, and were widely read. This understanding of fundamentals lead to a quick succession of inventions —the wireless telegraph (1899), the vacuum tube (1905), and radio broadcasting —that marked the true emergence of the communications age.

During the first part of the twentieth century, circuit theory and electromagnetic theory were all an electrical engineer needed to know to be qualified and produce first-rate designs. Consequently, on the premise that what is important should be taught "early and often," circuit theory served as the foundation and the framework of all of electrical engineering education. At mid-century, three "inventions" changed the ground rules. These were the first public demonstration of the first electronic computer (1946), the invention of the transistor (1947), and the publication of *A Mathematical Theory of Communication* by Claude Shannon (1948). Although conceived separately, these creations gave birth to the information age, in which digital and analog communication systems interact and compete for design preferences. About twenty years later, the laser was invented, which opened even more design possibilities. Thus, the primary focus shifted from *how* to build communication systems (the circuit theory era) to *what* communications systems were intended to accomplish. Only once the intended system is specified can an implementation be selected. Today's electrical engineer must be mindful of the system's ultimate goal, and understand the tradeoffs between digital and analog alternatives, and between hardware and software configurations in designing information systems.

1.2 What is Information?

Information is something that you are interested in. It could be a song, a football score, a picture of your family, stock quotes . . . Engineers, who don't care about information content, categorize information into two fundamental forms: continuous-valued or discrete-valued. Examples of the former is a song or human speech, where information is expressed by variations in air pressure, and images, where color and darkness correspond to variations in the surface properties of a sheet of paper or the emissions of a computer screen. We use the term *analog* information to denote all such continuous-valued quantities. An example of discrete-valued information is letters of the alphabet, and the sports score and stock quote examples elaborate this information type category. Such information is said to be *digital* information.

Whether analog or digital, information is represented by the fundamental quantity in electrical engineering: the *signal*. Stated in mathematical terms, *a signal is merely a function*. Analog signals are continuous-valued; digital ones discrete-valued. The independent variable of the signal could be time (speech, for example), space (images), or the integers (denoting the sequencing of letters and numbers in the football score).

1.2.1 Analog Information

Analog signals are usually signals defined over continuous independent variable(s). Speech, which we discuss in some detail, is produced by your vocal cords exciting acoustic resonances in your vocal tract. The result is pressure waves propagating in the air, and the speech signal thus corresponds to a function having independent variables of space and time and a value corresponding to air pressure: $s(\mathbf{x}, t)$.[*] Photographs are static, and are continuous-valued signals defined over space. Black-and-white images have only one value at each point in space, which amounts to

---

[*] Here we use vector notation $\mathbf{x}$ to denote spatial coordinates.

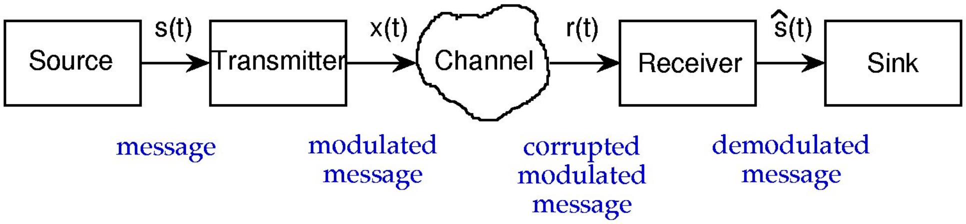


Figure 1.1 The Fundamental Model of Communication.

its optical reflection properties. Color images have values that express how reflectivity depends on the optical spectrum. Painters long ago found that mixing together combinations of the so-called primary colors —red, yellow and blue —can produce very realistic color images. Thus, images today are usually thought of as having three values at every point in space, but a different set of colors is used: How much of red, *green* and blue is present. Mathematically, color pictures are multivalued—vector-valued—signals: $\mathbf{s}(\mathbf{x}) = [r(\mathbf{x})\, g(\mathbf{x})\, b(\mathbf{x})]$.

Interesting cases abound where the analog signal depends not on a continuous variable, such as time, but on a discrete values. For example, temperature readings taken every hour have continuous—analog—values, but the signal's independent variable is (essentially) the integers.

1.2.2 Digital Information

The word "digital" means discrete-valued and implies the signal has an integer-valued independent variable. Digital information includes numbers and symbols (characters typed on the keyboard, for example). Computers rely on the digital representation of information to manipulate and transform information. Symbols do not have a numeric value, and each is represented by unique number. The ASCII character code has the upper- and lowercase characters, the numbers, punctuation marks, and various other symbols represented by a seven-bit integer. For example, the ASCII code represents the letter a as the number 97_{10} and the letter A as 65_{10}. Table 1.1 shows the international convention on associating characters with integers.

1.3 Communication Systems

The fundamental model of communications is portrayed in Fig. 1.1. In the fundamental system, each message-bearing signal $s(t)$ is analog and is a function of time. A *system* operates on zero, one, or several signals to produce more signals

00	nul	01	soh	02	stx	03	etx	04	eot	05	enq	06	ack	07	bel
08	bs	09	ht	0A	nl	0B	vt	0C	np	0D	cr	0E	so	0F	si
10	dle	11	dc1	12	dc2	13	dc3	14	dc4	15	nak	16	syn	17	etb
18	can	19	em	1A	sub	1B	esc	1C	fs	1D	gs	1E	rs	1F	us
20	sp	21	!	22	"	23	#	24	$	25	%	26	&	27	’
28	(	29	)	2A	*	2B	+	2C	,	2D	−	2E	.	2F	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3A	:	3B		3C	<	3D	=	3E	>	3F	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4A	J	4B	K	4C	L	4D	M	4E	N	4F	O
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5A	Z	5B	[	5C	\	5D	]	5E	^	5F	_
60	‘	61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h	69	i	6A	j	6B	k	6C	l	6D	m	6E	n	6F	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7A	z	7B	{	7C	|	7D	}	7E	~	7F	del

Table 1.1 The ASCII translation table shows how standard keyboard characters are represented by integers. This table displays the so-called 7-bit code (how many characters in a seven-bit code?); extended ASCII has an 8-bit code. The numeric codes are represented in hexadecimal (base-16) notation. The mnemonic characters correspond to control characters, some of which may be familiar (like `cr` for carriage return) and some not (`bel` means a "bell").

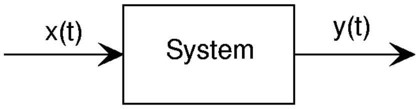


Figure 1.2 The system depicted here has input $x(t)$ and output $y(t)$. Mathematically, systems operate on function(s) to produce other function(s). In many ways, systems are like functions, rules that yield a value for the dependent variable (our output signal) for each value of its independent variable (its input signal). The notation $y(t) = \mathcal{S}[x(t)]$ corresponds to this block diagram. We term $\mathcal{S}[\cdot]$ the *input-output relation* for the system.

or to simply absorb them (Figure 1.2 {5}). In electrical engineering, we represent a system as a box, receiving input signals (usually coming from the left) and producing from them new output signals. This graphical representation is known as a *block diagram.* We denote input signals by lines having arrows pointing into the box, output signals by arrows pointing away. As typified by the communications model, how information flows, how it is corrupted and manipulated, and how it is ultimately received is summarized by interconnecting block diagrams: The outputs of one of more systems serve as the inputs to others.

In the communications model, the *source* produces a signal that will be absorbed by the *sink.* Examples of time-domain signals produced by a source are music, speech, and characters typed on a keyboard. Signals can also be functions of two variables — an image is a signal that depends on two spatial variables —or more — television pictures (video signals) are functions of two spatial variables and time. Thus, information sources produce signals. *In physical systems, each signal corresponds to an electrical voltage or current.* To be able to design systems, we must understand electrical science and technology. However, we first need to understand the big picture to appreciate the context in which the electrical engineer works.

In communication systems, messages — signals produced by sources —must be recast for transmission. The block diagram has the message $s(t)$ passing through a block labeled *transmitter* that produces the signal $x(t)$. In the case of a radio transmitter, it accepts an input audio signal and produces a signal that physically is an electromagnetic wave radiated by an antenna and propagating as Maxwell's equations predict. In the case of a computer network, typed characters are encapsulated in packets, attached with a destination address, and launched into the Internet. From the communication systems "big picture" perspective, the *same* block diagram applies although the systems can be very different. In any case, the transmitter should *not* operate in such a way that the message $s(t)$ cannot be recovered from $x(t)$. In the mathematical sense, the inverse system must exist, else the communication system cannot be considered reliable.*

Transmitted signals next pass through the next stage, the evil *channel.* Nothing good happens to a signal in a channel: It can become corrupted by noise, distorted, and attenuated among many possibilities. The channel cannot be escaped (the real world is cruel), and transmitter design *and* receiver design focus on how best to jointly fend off the channel's effects on signals. The channel is another system in our block diagram, and produces $r(t)$, the signal *received* by the receiver. If the channel were benign (good luck finding such a channel in the real world), the receiver would serve as the inverse system to the transmitter, and yield the message with no distortion. However, because of the channel, the receiver must do its best to produce a received message $\hat{s}(t)$ that resembles $s(t)$ as much as possible. Shannon showed in his 1948 paper was that reliable —for the moment, take this word to mean error-free — digital communication was possible over arbitrarily noisy channels. It is this result that modern communications systems exploit, and why many communications systems are going "digital." Chapter 6 details Shannon's theory

---

*It is ridiculous to transmit a signal in such a way that *no one* can recover the original. However, clever systems exist that transmit signals so that only the "in crowd" can recover them. Such crytographic systems underly secret communications.

of information, and there we learn of Shannon's result and how to use it.

Finally, the received message is passed to the information *sink* that somehow makes use of the message. In the communications model, the source is a system having no input but producing an output; a sink has an input and no output.

Understanding signal generation and how systems work amounts to understanding signals, the nature of the information they represent, how information is transformed between analog and digital forms, and how information can be processed by systems operating on information-bearing signals. This understanding demands two different fields of knowledge. One is electrical science: How are signals represented and manipulated electrically? The second is signal science: What is the structure of signals, no matter what their source, what is their information content, and what capabilities does this structure force upon communication systems?

1.4 The Fundamental Signal

The most ubiquitous signal in electrical engineering is the *sinusoid*.

$$\boxed{s(t) = A\cos(2\pi f t + \phi) \quad \text{or} \quad A\cos(\omega t + \phi)} \tag{1.1}$$

A is known as the sinusoid's *amplitude*, and determines the sinusoid's size. The amplitude conveys the sinusoid's physical units (volts, lumens, etc). The *frequency* f has units of Hz (Hertz) or s^{-1}, and determines how rapidly the sinusoid oscillates per unit time. The temporal variable t always has units of seconds, and thus the frequency determines how many oscillations/second the sinusoid has. AM radio stations have carrier frequencies of about 1 MHz (one mega-hertz or 10^6 Hz), while FM stations have carrier frequencies of about 100 MHz. Frequency can also be expressed by the symbol ω, which has units of radians/second. Clearly, $\omega = f/2\pi$. In communications, we most often express frequency in Hertz. Finally, ϕ is the phase, and determines the sine wave's behavior at the origin ($t = 0$). It has units of radians, but we can express it in degrees, realizing that in computations we must convert from degrees to radians. Note that if $\phi = -\pi/2$, the sinusoid corresponds to a sine function, having a zero value at the origin.

$$A\sin(2\pi f t + \phi) = A\cos(2\pi f t + \phi - \pi/2)\ .$$

Thus, the only difference between a sine and cosine signal is the phase; we term either a sinusoid.

We can also define a discrete-time variant of the sinusoid: $A\cos(2\pi f n + \phi)$. Here, the independent variable is n and represents the integers. Frequency now has no dimensions, and takes on values between 0 and 1.

Question 1.1 $\{247\}$ *Show that* $\cos 2\pi f n = \cos 2\pi (f+1)n$*, which means that a sinusoid having a frequency larger than one corresponds to a sinusoid having a frequency less than one.*

Notice that we shall call either sinusoid an analog signal. Only when the discrete-time signal takes on a finite set of values can it be considered a digital signal.

Question 1.2 $\{247\}$ *Can you think of a simple signal that has a finite number of values but is defined in continuous time? Such a signal is also an analog signal.*

1.5 Communicating Information with Signals

The basic idea of communication engineering is to use a signal's parameters to represent either real numbers or other signals. The technical term is to *modulate* the *carrier* signal's parameters to transmit information from one place to another. To explore the notion of modulation, we can send a real number (today's temperature, for example) by changing a sinusoid's amplitude accordingly. If we wanted to send the daily temperature, we would keep the frequency constant (so the receiver would know what to expect) and change the amplitude at midnight. We could relate temperature to amplitude by the formula $A = A_0 \cdot (1 + kT)$, where A_0 and k are constants that the transmitter and receiver must *both* know.

If we had two numbers we wanted to send at the same time, we could modulate the sinusoid's frequency as well as its amplitude.* This modulation scheme assumes we can estimate the sinusoid's amplitude and frequency; we shall learn that this is indeed possible.

Now suppose we have a sequence of parameters to send. We have exploited all of the sinusoid's two parameters. What we can do is modulate them for a limited time (say T seconds), and send two parameters every T. This simple notion corresponds to how a modem works. Here, typed characters are encoded into eight bits, and the individual bits are encoded into a sinusoid's amplitude and frequency. We'll learn how this is done in subsequent chapters, and more importantly, we'll learn what the limits are on such digital communication schemes.

---

*Modulating the phase is trickier because phase modulation requires time synchronization, a frequently unavailable option.

Problems

1.1 RMS Values

The *rms* (root-mean-square) value of a periodic signal is defined to be

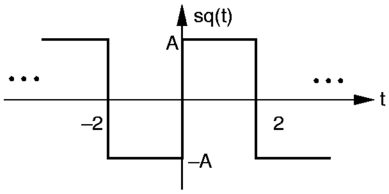
$$\langle s \rangle = \sqrt{\frac{1}{T}\int_0^T s^2(t)\,dt}\ ,$$

where T is defined to be the signal's *period*: the smallest positive number such that $s(t) = s(t+T)$.

(a) What is the period of $s(t) = A\sin(2\pi f_0 t + \phi)$?

(b) What is the rms value of this signal? How is it related to the peak value?

(c) What is the period and rms value of the depicted *square wave*, generically denoted by $\text{sq}(t)$?



(d) By inspecting any device you plug into a wall socket, you'll see that it is labeled "110 volts AC." What is the expression for the voltage provided by a wall socket? What is its rms value?

1.2 Modems

The word "modem" is short for "modulator-demodulator." Modems are used not only for connecting computers to telephone lines, but also for connecting digital (discrete-valued) sources to generic channels. In this problem, we explore a simple kind of modem. To transmit symbols, such as letters of the alphabet, RU computer modems use two frequencies (1600 and 1800 Hz) and several amplitude levels. A transmission is sent for a period of time T (known as the transmission or baud interval) and equals the sum of these amplitude-weighted carriers.

$$x(t) = A_1\sin(2\pi f_1 t) + A_2\sin(2\pi f_2 t) \quad 0 \le t \le T$$

We send successive symbols by choosing the appropriate frequency and amplitude combinations, and sending them one after another.

(a) What is the smallest transmission interval that makes sense to use with the frequencies given above? In other words, what should T be so that an integer number of cycles of the carrier occurs?

(b) Sketch (using Matlab) the signal that modem produces over several transmission intervals. Make use your axes are labeled.

We use a discrete set of values for A_1 and A_2. If we have N_1 values for amplitude A_1, and N_2 values for A_2, we have $N_1 N_2$ possible symbols that can be sent during each T second interval. To convert this number into bits (the fundamental unit information engineers use to quantify things), compute $\log_2 N_1 N_2$.

(c) Using your symbol transmission interval, how many amplitude levels are needed to transmit ASCII characters at a datarate of 3,200 bits/s? Assume use of the extended (8-bit) ASCII code.

(d) The classic communications block diagram applies to the modem. Discuss how the transmitter must interface with the message source since the source is producing letters of the alphabet, not bits.

Chapter 2

Signals and Systems

2.1 Signal Theory

Elemental signals are the building blocks with which we build complicated signals. By definition, elemental signals have a simple structure. Exactly what we mean by the "structure of a signal" will unfold in this chapter. Signals are nothing more than functions defined with respect to some independent variable, which we take to be time for the most part. Very interesting signals are not functions solely of time; one great example is images. For them, the independent variables are x and y (two-dimensional space). Video signals are functions of three variables: two spatial dimensions and time. Most of the ideas underlying modern signal theory can be exemplified with one-dimensional signals.

Sinusoids. Perhaps the most common real-valued signal is the sinusoid.

$$s(t) = A\cos(2\pi f_0 t + \phi)$$

For this signal, A is its amplitude, f_0 its frequency, and ϕ its phase.

Complex exponentials. The most important signal is complex-valued, the complex exponential.

$$\begin{aligned} s(t) &= Ae^{j(2\pi f_0 t + \phi)} \\ &= Se^{j2\pi f_0 t}, \quad S = Ae^{j\phi} \end{aligned}$$

Here, j denotes $\sqrt{-1}$* and S is known as the signal's *complex amplitude.* Considering the complex amplitude as a complex number in polar form, its magnitude is the amplitude A and its angle the signal phase. The complex amplitude is also known as a *phasor*. The complex exponential cannot be further decomposed into more elemental signals, and is the *most important signal in electrical engineering*! Mathematical manipulations at first appear to be more difficult because complex-valued numbers are introduced. In fact, early in the twentieth century, mathematicians thought engineers would not be sufficiently sophisticated to handle complex exponentials even though they greatly simplified solving circuit problems. Steinmetz introduced complex exponentials to electrical engineering, and demonstrated that "mere" engineers could use them to good effect and even obtain right answers! See Appendix A for a review of complex numbers and complex arithmetic.

---

*Mathematicians use the symbol i for $\sqrt{-1}$. For electrical engineers, history led to a different symbol. Ampère used the symbol i to denote current (intensité de current). Euler first used i for

The complex exponential defines the notion of frequency: It is the *only* signal that contains only one frequency component. The sinusoid consists of two frequency components: one at the frequency $+f_0$ and the other at $-f_0$. This decomposition of the sinusoid can be traced to Euler's relation.

$$\cos(2\pi ft) = \frac{e^{j2\pi ft} + e^{-j2\pi ft}}{2} \tag{2.1}$$

$$\sin(2\pi ft) = \frac{e^{j2\pi ft} - e^{-j2\pi ft}}{2j} \tag{2.2}$$

$$e^{j2\pi ft} = \cos(2\pi ft) + j\sin(2\pi ft) \tag{2.3}$$

The complex exponential signal can thus be written in terms of its real and imaginary parts using Euler's relation. Thus, sinusoidal signals can be expressed as either the real or the imaginary part of a complex exponential signal, the choice depending on whether cosine or sine phase is needed, or as the sum of two complex exponentials. These two decompositions are mathematically equivalent to each other.

$$\begin{aligned} A\cos(2\pi ft + \phi) &= \mathrm{Re}\left[Ae^{j\phi}e^{j2\pi ft}\right] \\ A\sin(2\pi ft + \phi) &= \mathrm{Im}\left[Ae^{j\phi}e^{j2\pi ft}\right] \end{aligned}$$

Using the complex plane, we can envision the complex exponential's temporal variations (Fig. 2.1). The magnitude of the complex exponential is A, and the initial value of the complex exponential at $t = 0$ has an angle of ϕ. As time increases, the locus of points traced by the complex exponential is a circle (it has constant magnitude of A). The number of times per second we go around the circle equals the frequency f. The time taken for the complex exponential to go around the circle

---

imaginary numbers but that notation did not take hold until roughly Ampère's time. It wasn't until the twentieth century that the importance of complex numbers to circuit theory became evident. By then, i for current was entrenched and electrical engineers chose j for writing complex numbers.

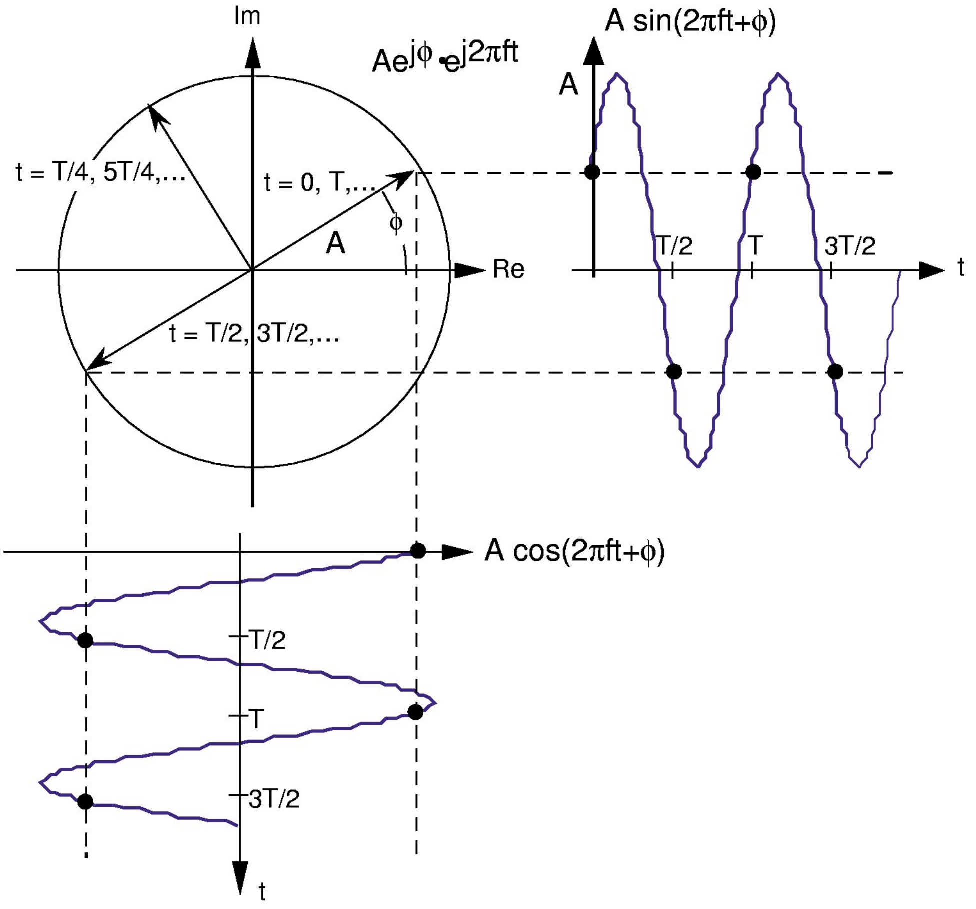

Im
Aejϕ •ej2πft
t = T/4, 5T/4,...
t = 0, T,...
ϕ
A
Re
t = T/2, 3T/2,...
A sin(2πft+ϕ)
A
T/2
T
3T/2
t
A cos(2πft+ϕ)
T/2
T
3T/2
t

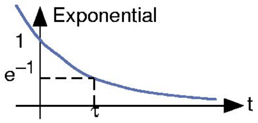
Figure 2.1 Graphically, the complex exponential scribes a circle in the complex plane as time evolves. Its real and imaginary parts are sinusoids. The rate at which the signal goes around the circle is the frequency f and the time taken to go around is the *period* T. A fundamental relationship is $T = 1/f$.

---

once is known as its *period* T, and equals $1/f$. The projections onto the real and imaginary axes of the rotating vector representing the complex exponential signal are the cosine and sine signal of Euler's relation (2.1).

Real exponentials. As opposed to complex exponentials which oscillate, real exponentials decay.

$$s(t) = e^{-t/\tau}$$



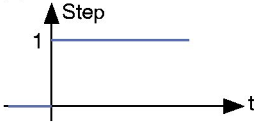
The quantity τ is known as the exponential's *time constant*, and corresponds to the time required for the exponential to decrease by a factor of $1/e$, which equals 0.368. A decaying complex exponential is the product of a real and a complex exponential.

$$s(t) = Se^{-t/\tau}e^{j2\pi ft} = Se^{(-1/\tau+j2\pi f)t}$$

In the complex plane, this signal corresponds to an exponential spiral. For such signals, we can define *complex frequency* as the quantity multiplying t.

Unit step. The step function is denoted by $\mathrm{u}(t)$, and is defined to be

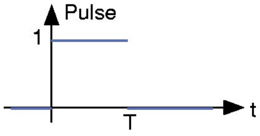
$$\mathrm{u}(t) = \begin{cases} 0, & t < 0 \\ 1, & t > 0 \end{cases}$$



This signal is discontinuous at the origin. Its value at the origin need not be defined, and doesn't matter in signal theory. This kind of signal is used to describe signals that "turn on" suddenly. For example, to mathematically represent turning on an oscillator, we can write it as the product of a sinusoid and a step: $s(t) = A\sin(2\pi ft)\,\mathrm{u}(t)$.

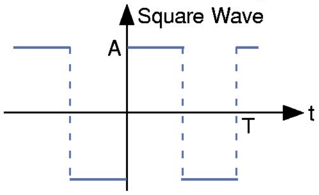
Pulse. The unit pulse describes turning a unit-amplitude signal on for a duration of Δ seconds, then turning it off.

$$\mathrm{p}_\Delta(t) = \begin{cases} 0, & t < 0 \\ 1, & 0 < t < \Delta \\ 0, & t > \Delta \end{cases}$$



We will find that this is the second most important signal in communications.

Square wave. The square wave $\mathrm{sq}(t)$ is a periodic signal like the sinusoid. It too has an amplitude and a period, which must be specified to characterize the signal. We find subsequently that the sine wave is a simpler signal than the square wave.



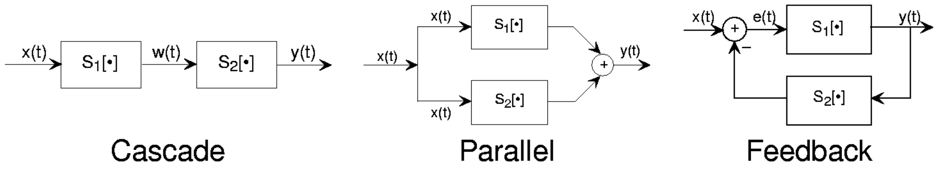


Figure 2.2 The most rudimentary ways of interconnecting systems are the cascade (left), parallel (center), and feedback (right) configurations.

2.2 System Theory

Signals are manipulated by systems. As shown in Figure 1.2 $\{5\}$, systems that process information have an input signal (or have several input signals), and produce output signal(s). Mathematically, we represent what a system does by the notation $y(t) = \mathcal{S}\left[x(t)\right]$, with x representing the input signal and y the output. This notation mimics the mathematical symbology of a function: A system's input is analogous to an independent variable and its output the dependent variable. For the mathematically inclined, a system is a *functional*: a function of a function (signals are functions)

2.2.1 Connecting Systems Together

Simple systems can be connected together—one system's output becomes another's input—to accomplish some overall design. Interconnection topologies can be quite complicated, but usually consist of weaves of three basic interconnection forms.

Cascade interconnection. The simplest form is one system's output is connected only to another's input. Mathematically, $w(t) = \mathcal{S}_1\left[x(t)\right]$, and $y = \mathcal{S}_2\left[w(t)\right]$, with the information contained in $x(t)$ processed by the first, then the second system. In some cases, the ordering of the system's matter, in others it does not. For example, in the fundamental communications model (Figure 1.1 $\{4\}$) the ordering most certainly matters.

Parallel interconnection. A signal $x(t)$ is routed to two (or more) systems, with this signal appearing as the input to all systems simultaneously and with equal strength.* Two or more systems operate on $x(t)$ and their outputs are added together to create the output $y(t)$. Thus, $y(t) = \mathcal{S}_1\left[x(t)\right] + \mathcal{S}_2\left[x(t)\right]$, and the information in $x(t)$ is processed separately by both systems.

---

*Block diagrams have the convention that signals going to more than one system are not split into pieces along the way.

Feedback interconnection. The subtlest interconnection configuration has a system's output also contributing to its input. Engineers would say the output is "fed back" to the input through system 2, hence the terminology. The mathematical statement of the feedback interconnection is that the feed-forward system produces the output: $y(t) = \mathcal{S}_1[e(t)]$. Its input equals the input signal minus the output of some other system's output to $y(t)$: $e(t) = x(t) - \mathcal{S}_2[y(t)]$. Feedback systems are omnipresent in control problems, with the error signal used to adjust the output to achieve some condition defined by the input (controlling) signal. For example, in a car's cruise control system, $x(t)$ is a constant representing what speed you want, and $y(t)$ the car's speed as measured by a speedometer. In this application, system 2 is the identity system (output equals input).

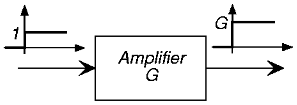
2.2.2 Simple Systems

Systems manipulate signals, creating output signals derived from their inputs. Why the following are categorized as "simple" will only become evident once the book is completed.

Sources. Sources produce signals without having input. We like to think of these as having controllable parameters, like amplitude and frequency. Examples would be oscillators that produce periodic signals like sinusoids and square waves and noise generators that yield signals erratic waveforms (more about noise subsequently). Simply writing an expression for the signals they produce specifies sources. A sine wave generator might be specified by $y(t) = A\sin(2\pi f_0 t)\,\mathrm{u}(t)$, which says that the source was turned on at $t = 0$ to produce a sinusoid of amplitude A and frequency f_0.

Amplifiers. An amplifier multiplies its input by a constant known as the amplifier *gain.*

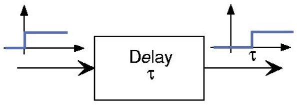
$$y(t) = Gx(t)$$



The gain can be positive or negative (if negative, we would say that the amplifier *inverts* its input) and can be greater than one or less than one. If less than one, the amplifier actually *attenuates*. A real-world example of an amplifier is your home stereo. You control the gain by turning the volume control.

Delay. A system serves as a time delay when the output signal equals the input signal at an earlier time.

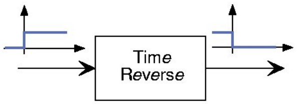
$$y(t) = x(t - \tau)$$



Here, τ is the delay. The way to understand this system is to focus on the time origin: The output at time $t = \tau$ equals the input at time $t = 0$. Thus, if the delay is positive, the output emerges later than the input, and plotting the output amounts to shifting the input plot to the right. The delay can be negative, in which case we say the system *advances* its input. Such systems are difficult to build (they would have to produce signal values derived from what the input *will be*), but we will have occasion to advance signals in time.

Time reversal. Here, the output signal equals the input signal flipped about the time origin.

$$y(t) = x(-t)$$



Again, such systems are difficult to build, but the notion of time reversal occurs frequently in communications systems.

Question 2.1 $\{$**247**$\}$ *Mentioned earlier was the issue of whether the ordering of systems mattered. In other words, if we have two systems in cascade, does the output depend on which comes first? Determine if the ordering matters for the cascade of an amplifier and a delay and for the cascade of a time-reversal system and a delay.*

Derivative systems and integrators. Systems that perform calculus-like operations on their inputs can produce waveforms significantly different than present in the input. Derivative systems operate in a straightforward way: A first-derivative system would have the input-output relationship $y(t) = \frac{dx(t)}{dt}$. Integral systems have the complication that the integral's limits must be defined. It is a signal theory convention that the elementary integral operation have a lower limit of $-\infty$, and that the value of *all* signals at $t = -\infty$ equals zero. A simple integrator would have input-output relation

$$y(t) = \int_{-\infty}^{t} x(\alpha)\, d\alpha \,.$$

Linear systems. Linear systems are a *class* of systems rather than having a specific input-output relation. Linear systems form the foundation of system theory, and are the most important class of systems in communications. They have the property that when the input is expressed as a weighted sum of component signals, the output equals the same weighted sum of the outputs produced by each component. When $\mathcal{S}[\cdot]$ is linear,

$$\boxed{\mathcal{S}[G_1x_1(t)+G_2x_2(t)] = G_1\mathcal{S}[x_1(t)]+G_2\mathcal{S}[x_2(t)]}$$

for all choices of signals and gains.

This general input-output relation property can be manipulated to indicate specific properties shared by all linear systems.

- $\mathcal{S}[Gx(t)] = G\mathcal{S}[x(t)]$
  The colloquialism summarizing this property is "Double the input, you double the output." Note that this property is consistent with alternate ways of expressing gain changes: Since $2x(t)$ also equals $x(t)+x(t)$, the linear system definition provides the same output no matter which of these is used to express a given signal.
- $\mathcal{S}[0] = 0$
  If the input is *identically zero for all time*, the output of a linear system must be zero. This property follows from the simple derivation $\mathcal{S}[0] = \mathcal{S}[x(t)-x(t)] = \mathcal{S}[x(t)] - \mathcal{S}[x(t)] = 0$.

Just why linear systems are so important is related not only to their properties, which are divulged throughout this book, but also because they lend themselves to relatively simple mathematical analysis. Said another way, "They're the only systems we thoroughly understand!"

Time-invariant systems. Systems that don't change their input-output relation with time are said to be time-invariant. The mathematical way of stating this property is to use the signal delay concept described previously $\{16\}$.

$$\text{If } y(t) = \mathcal{S}[x(t)],\ y(t-\tau) = \mathcal{S}[x(t-\tau)]\ \forall\tau$$

If you delay (or advance) the input, the output is similarly delayed (advanced). Thus, a time-invariant system responds to an input you may supply tomorrow the same way it responds to the same input applied today; today's output is merely delayed to occur tomorrow.

The collection of linear, time-invariant systems are *the* most thoroughly understood systems. Much of the signal processing and system theory discussed here concentrates on such systems. For example, electric circuits are, for the most part, linear and time-invariant. Nonlinear ones abound, but characterizing them so that you can predict their behavior for any input remains an unsolved problem.

Input-Output Relation	*Linear?*	*Time-Invariant?*
$y(t) = \frac{dx}{dt}$	$\surd$	$\surd$
$y(t) = \frac{d^2x}{dt^2}$	$\surd$	$\surd$
$y(t) = \left(\frac{dx}{dt}\right)^2$	$\times$	$\surd$
$y(t) = \frac{dx}{dt} + x$	$\surd$	$\surd$
$y(t) = x_1(t) + x_2(t)$	$\surd$	$\surd$
$y(t) = x(t-\tau)$	$\surd$	$\surd$
$y(t) = \cos(2\pi ft) \cdot x(t)$	$\surd$	$\times$
$y(t) = x(-t)$	$\surd$	$\times$
$y(t) = x^2(t)$	$\times$	$\surd$
$y(t) = \lvert x(t) \rvert$	$\times$	$\surd$
$y(t) = mx(t) + b$	$\times$	$\surd$

2.3 Signal Decomposition

A signal's complexity is not related to how wiggly it is. Rather, a signal expert looks for ways of decomposing a given signal into a *sum of simpler signals*, which we term the *signal superposition*. Though we will never compute a signal's complexity, it essentially equals the number of terms in its decomposition. In writing a signal as a sum of component ones, we can change the component signal's gain by multiplying it by a constant and by delaying it. More complicated decompositions could contain derivatives or integrals of simple signals. In short, signal decomposition amounts to thinking of the signal as the output of a linear system having simple signals as its inputs. We would build such a system, but envisioning the signal's components helps understand the signal's structure. Furthermore, you can readily compute a linear system's output to an input decomposed as a superposition of simple signals.

Example

As an example of signal complexity, we can express the pulse $\mathrm{p}_\Delta(t)$ as a sum of delayed unit steps.

$$\mathrm{p}_\Delta(t) = \mathrm{u}(t) - \mathrm{u}(t-\Delta)$$

Thus, the pulse is a more complex signal than the step. Be that as it may, the pulse is very useful to us.

Question 2.2 $\{$**247**$\}$ *Express a square wave having period* T *and amplitude* A *as a superposition of delayed and amplitude-scaled pulses.*

Because the sinusoid is a superposition of two complex exponentials, the sinusoid is more complex.* The complex exponential can also be written (using Euler's relation $\{12\}$) as a sum of a sine and a cosine. We will discover that virtually every signal can be decomposed into a sum of complex exponentials, and that this decomposition is *very* useful. Thus, the complex exponential is more fundamental, and Euler's relation does not adequately reveal its complexity.

Problems

2.1 Complex-valued Signals

Complex numbers and phasors play a very important role in electrical engineering. Solving systems for complex exponentials is much easier than for sinusoids, and linear systems analysis is particularly easy.

(a) Find the phasor representation for each, and re-express each as the real and imaginary parts of a phasor. What is the frequency (in Hz) of each? In general, are your answers unique? If so, prove it; if not, find an alternative answer for the phasor representation.

(i) $3\sin(24t)$ (ii) $\sqrt{2}\cos(2\pi 60t+\pi/4)$

(iii) $2\cos(t+\pi/6)+4\sin(t-\pi/3)$

(b) Show that for linear systems having real-valued outputs for real inputs, that when the input is the real part of a phasor, the output is real part of the system's output to the phasor.

$$\mathcal{S}\left[\mathrm{Re}\left[Ae^{j2\pi ft}\right]\right]=\mathrm{Re}\left[A\mathcal{S}\left[e^{j2\pi ft}\right]\right]$$

2.2 For each of the indicated voltages, write it as the real part of a complex exponential ($v(t)=\mathrm{Re}\left[Ve^{st}\right]$). Explicitly indicate the value of the complex amplitude V and the complex frequency s. Represent each complex amplitude as a vector in the V-plane, and indicate the location of the frequencies in the complex s-plane.

(a) $v(t)=\cos 5t$ **(b)** $v(t)=\sin(8t+\pi/4)$

(c) $v(t)=e^{-t}$ **(d)** $v(t)=e^{-3t}\sin(4t+3\pi/4)$

(e) $v(t)=5e^{2t}\sin(8t+2\pi)$ **(f)** $v(t)=-2$

(g) $v(t)=4\sin 2t+3\cos 2t$

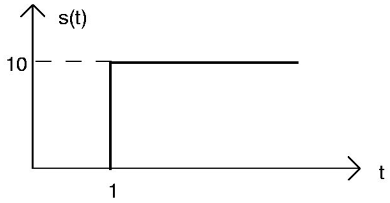
(h) $v(t)=2\cos(100\pi t+\pi/6)-\sqrt{3}\sin(100\pi t+\pi/2)$

---

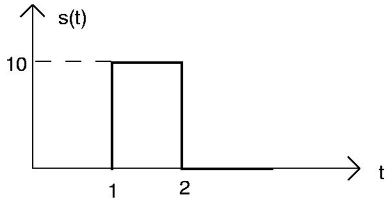
*We could not prevent ourselves from the pun in this statement. Clearly, the word "complex" is used in two different ways here.

2.3 Express each of the following signals as a linear combination of delayed and weighted step functions and *ramps* (the integral of a step).

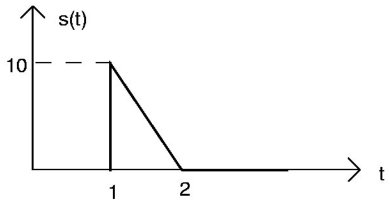
(a)



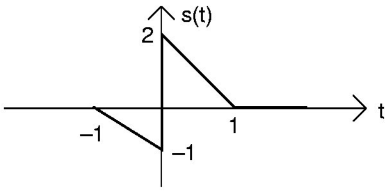
(b)



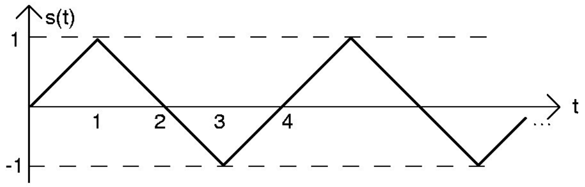
(c)



(d)

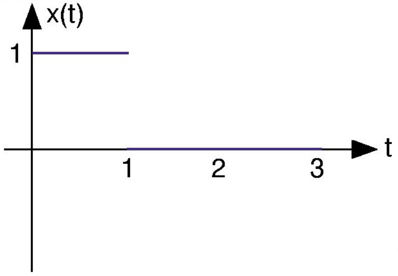


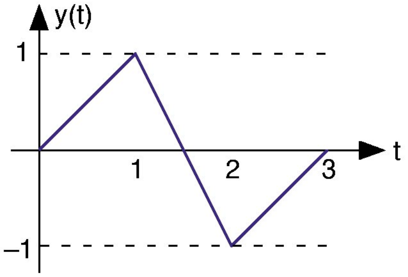
(e)



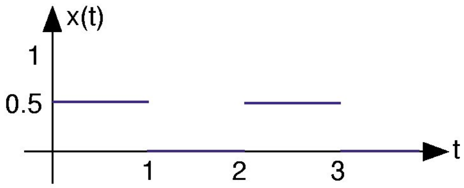
2.4 **Linear, Time-Invariant Systems**

When the input to a linear, time-invariant system is the signal $x(t)$, the output is the signal $y(t)$.





(a) Find and sketch this system's output when the input is the signal depicted below.



(b) Find and sketch this system's output when the input is a unit step.

Chapter 3

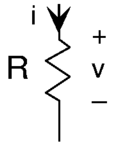
Analog Signal Processing

We know that information can be represented by signals; now we need to understand how signals are physically realized. Over the years, electric signals have been found to be the easiest to use. Voltage and currents comprise the electric instantiations of signals. Thus, we need to delve into the world of electricity and electromagnetism. The systems used to manipulate electric signals directly are called *circuits*, and they refine the information representation or extract information from the voltage or current. In many cases, they make nice examples of linear systems.

3.1 Voltage and Current

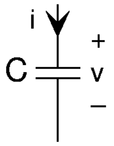
A generic circuit element places a constraint between the classic variables of a circuit: voltage and current. *Voltage* is electric potential and represents the "push" that drives electric charge from one place to another. What causes charge to move is a physical separation between positive and negative charge. A battery generates, through electrochemical means, excess positive charge at one terminal and negative charge at the other, creating an electric field. Voltage is defined *across* a circuit element, with the positive sign denoting a positive voltage drop across the element. When a conductor connects the positive and negative potentials, *current* flows, with positive current indicating that positive charge folows from the positive terminal to the negative. Electrons comprise current flow in many cases. Because electrons have a negative charge, electrons move in the opposite direction of positive current flow: Negative charge flowing to the right is equivalent to positive charge moving to the left.

It is important to understand the physics of current flow in conductors to appreciate the innovation of new electronic devices. Electric charge can arise from many



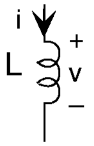
Resistor

$v = Ri$



Capacitor

$i = C\frac{dv}{dt}$



Inductor

$v = L\frac{di}{dt}$

sources, the simplest being the electron. When we say that "electrons flow through a conductor," what we mean is that the conductor's constituent atoms freely give up electrons from their outer shells. "Flow" thus means that electrons hop from atom to atom driven along by the applied electric potential. A missing electron, however, is a virtual positive charge. Electrical engineers call these *holes*, and in some materials, particularly certain semiconductors, current flow is actually due to holes. Current flow also occurs in nerve cells found in your brain. Here, neurons "communicate" using propagating voltage pulses that rely on the flow of positive ions (potassium and sodium primarily, and to some degree calcium) across the neuron's outer wall. Thus, current can come from many sources, and circuit theory can be used to understand how current flows in reaction to electric fields.

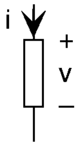


Figure 3.1 The generic circuit element.

Current flows through circuit elements, such as that depicted in Figure 3.1, and through conductors, which we indicate by lines in circuit diagrams. For every circuit element we define a voltage and a current. The element has a *v*-*i* relation defined by the element's physical properties. In defining the *v*-*i* relation, we have the convention that positive current flows from positive to negative voltage drop. Voltage has units of volts, and both the unit and the quantity are named for Volta. Current has units of amperes, and is named for the French physicist Ampère.

3.2 Linear Circuit Elements

The elementary circuit elements—the resistor, capacitor, and inductor—impose *linear* relationships between voltage and current.

Resistor. The resistor is far and away the simplest circuit element. Here, the voltage is proportional to the current, with the constant of proportionality R known as the *resistance*. Resistance has units of ohms, denoted by Ω, named for the German electrical scientist Georg Ohm. Sometimes, the *v*-*i* relation for the resistor is written $i = Gv$, with G, the *conductance*, equal to $1/R$. Conductance has units of Siemens (S), and is named for the German electronics industrialist Siemens.

As the resistance approaches infinity, we have what is known as an *open circuit*: No current flows but a non-zero voltage can appear across the open circuit. As the resistance becomes zero, the voltage goes to zero for a non-zero current flow. This situation corresponds to a *short circuit*. A superconductor physically realizes a short circuit.

Capacitor. The capacitor stores charge; as current is the rate of change of charge, integrating the capacitor's *v-i* relation yields $q = Cv$. The charge stored in a capacitor is proportional to the voltage. The constant of proportionality, the capacitance, has units of farads, and is named for the English experimental physicist Michael Faraday. If the voltage across a capacitor is constant, then the current flowing into it equals zero. In this situation, the capacitor is equivalent to an open circuit. The *v-i* relation can be expressed in integral form.

$$v(t) = \frac{1}{C} \int_{-\infty}^{t} i(\alpha)\, d\alpha$$

Inductor. The inductor stores magnetic flux, with larger valued inductors capable of storing more flux. Inductance has units of henries, and is named for the American physicist Joseph Henry. The integral form of the inductor's *v-i* relation is

$$i(t) = \frac{1}{L} \int_{-\infty}^{t} v(\alpha)\, d\alpha\ .$$

3.3 Sources

Sources of voltage and current are also circuit elements, but they are not linear in the strict sense of linear systems. For example, the voltage source's *v-i* relation is $v = v_s$ regardless what the current might be. As for the current source, $i = -i_s$ regardless of the voltage. Another name for a constant-valued voltage source is a battery, and can be purchased in any supermarket. Current sources, on the other hand, are much harder to find; we'll learn why later.

3.4 Ideal and Real-World Circuit Elements

Source and the linear circuit elements are *ideal* circuit elements. One central notion of circuit theory is combining the ideal elements to describe how physical elements

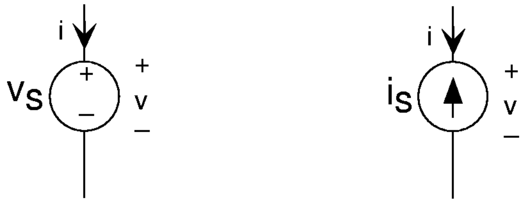


Figure 3.2 The voltage source on the left and current source on the right are like all circuit elements in that they have a particular relationship between the voltage and current defined for them. For the voltage source, $v = v_s$ for any current i; for the current source, $i = -i_s$ for any voltage v.

operate in the real world. For example, the 1 kΩ resistor you can hold in your hand is not exactly an ideal 1 kΩ resistor. First of all, physical devices are manufactured to close tolerances (the tighter the tolerance, the more money you pay), but never have exactly their advertised values. The fourth band on resistors specifies their tolerance; 10% is common. More pertinent to the current discussion is another deviation from the ideal: If a sinusoidal voltage is placed across a physical resistor, the current will not be exactly proportional to it as frequency becomes high, say above one MHz. At very high frequencies, the way the resistor is constructed introduces inductance and capacitance effects. Thus, the smart engineer must be aware of the frequency ranges over which their ideal models match well reality.

On the other hand, physical circuit elements can be readily found that well approximate the ideal, but they will always deviate from the ideal in some way. For example, a flashlight battery, like a C-cell, roughly corresponds a 1.5 v voltage source. However, it ceases to be modeled by a voltage source capable of supplying *any* current (that's what ideal ones can do!) when the resistance of the light bulb is too small.

3.5 Electric Circuits and Interconnection Laws

A *circuit* connects circuit elements together in a specific configuration designed to transform the source signal (originating from a voltage or current source) into another signal —the output —that corresponds to the current or voltage defined for a particular circuit element. A simple resistive circuit is shown in Figure 3.3 {27}. This circuit is the electrical embodiment of a system having its input provided by a source system producing $v_{\text{in}}(t)$.

To understand what this circuit accomplishes, we want to determine the voltage across the resistor labeled by its value R_2. Recasting this problem mathematically, we need to solve some set of equations so that we relate the output voltage v_{out}

to the source voltage. It would be simple —a little too simple at this point —if we could instantly write down the one equation that relates these two voltages. Until we have more knowledge about how circuits work, we must write a set of equations that allow us to find *all* the voltages and currents that can be defined for every circuit element. Because we have a three-element circuit, we have a total of six voltages and currents that must be either specified or determined. You can define the directions for current flow and positive voltage drop *any way you like*. When two people solve a circuit their own ways, the values of their variables may not agree, but current and voltage values for each element will agree. Do recall in defining your voltage and current variables {24} that the *v-i* relations for the elements presume that positive current flow is in the same direction as positive voltage drop. Once you define voltages and currents, we need six nonredundant equations to solve for the six unknown voltages and currents. By specifying the source, we have one; this amounts to providing the source's *v-i* relation. The *v-i* relations for the resistors give us two more. We are only halfway there.

What we need to solve every circuit problem is mathematical statements that express what the interconnection of elements is. Said another way, we need the laws that govern the electrical connection of circuit elements. First of all, the places where circuit elements attach to each other are called *nodes*. Two nodes are explicitly indicated in Figure 3.3 {27}; a third is at the bottom where the voltage source and resistor R_2 are connected. Electrical engineers tend to draw circuit

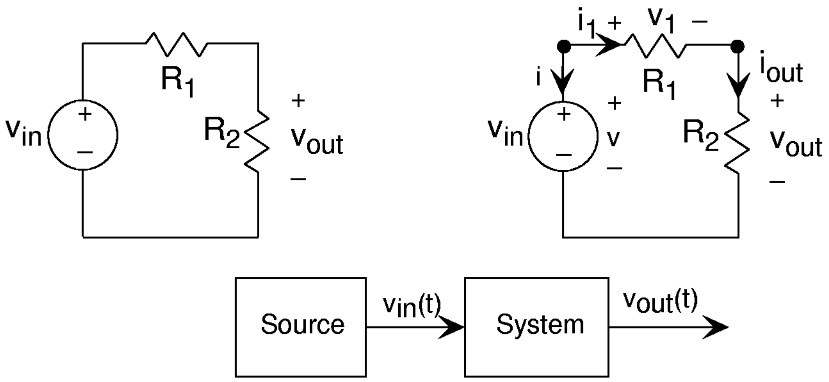


Figure 3.3 The circuit shown on the left is perhaps the simplest circuit that performs a signal processing function. Below it is the block diagram that corresponds to the circuit. The input is provided by the voltage source v_{in} and the output is the voltage v_{out} across the resistor label R_2. As shown on the right, we *analyze* the circuit —understand what it accomplishes — by defining currents and voltages for all circuit elements, and then solving the circuit and element equations.

diagrams — schematics — in a rectilinear fashion. Thus the long line connecting the bottom of the voltage source with the bottom of the resistor is intended to make the diagram look pretty. This line simply means that the two elements are connected together. *Kirchoff's Laws*, one for voltage and one for current, determine what a connection among circuit elements means.

Kirchoff's Current Law (KCL). At every node, the sum of all currents entering a node must equal zero. What this law means physically is that charge cannot accumulate in a node; what goes in must come out. We have a three-node circuit, and thus have three KCL equations.

$$-i - i_1 = 0 \qquad i_1 - i_2 = 0 \qquad i + i_2 = 0$$

Note that the current entering a node is the negative of the current leaving.

Given any two of these KCL equations, we can find the other by adding or subtracting them. Thus, one of them is redundant and, in mathematical terms, we can discard any one of them. The convention is to discard the one for the (unlabeled) node at the bottom of the circuit.

Question 3.1 $\{$**247**$\}$ *In writing KCL equations, you will find that in an n-node circuit, exactly one of them is* always *redundant. Can you sketch a proof of why this might be true? Hint: It has to do with the fact that charge won't accumulate in one place on its own.*

Kirchoff's Voltage Law (KVL). The voltage law says that the sum of voltages around every closed loop in the circuit must equal zero. A closed loop has the obvious definition: Starting at a node, trace a path through the circuit that returns you to the origin node. KVL expresses the fact that electric fields are conservative: The total work performed in moving a test charge around a closed path is zero. The KVL equation for our circuit is

$$v_1 + v_2 - v = 0\ .$$

In writing KVL equations, we follow the convention that an element's voltage enters with a plus sign if traversing the closed path, we go from the positive to the negative of the voltage's definition.

Returning to the example circuit, we have three *v*-*i* relations, two KCL equations, and one KVL equation for solving for the circuit's six voltages and currents.

$$\begin{array}{ll} & v = v_{\text{in}} \\ \textit{v-i}: & v_1 = R_1 i_1 \\ & v_{\text{out}} = R_2 i_{\text{out}} \\ \text{KCL}: & -i - i_1 = 0 \\ & i_1 - i_{\text{out}} = 0 \\ \text{KVL}: & -v + v_1 + v_{\text{out}} = 0 \end{array}$$

We have exactly the right number of equations! Eventually, we'll discover shortcuts for solving circuit problems; for now, we what to eliminate all variables but v_{out}. The KVL equation can be rewritten as $v_{\text{in}} = v_1 + v_{\text{out}}$. Substituting into it the resistor's *v-i* relations, we have $v_{\text{in}} = R_1 i_1 + R_2 i_{\text{out}}$.[*] One of the KCL equations says $i_1 = i_{\text{out}}$, which means that $v_{\text{in}} = R_1 i_{\text{out}} + R_2 i_{\text{out}} = (R_1 + R_2) i_{\text{out}}$. Solving for the current in the output resistor, we have $i_{\text{out}} = v_{\text{in}}/(R_1 + R_2)$. *We have now solved the circuit*: We have expressed one voltage or current in terms of sources and circuit-element values. To find any other circuit quantities, we can back substitute this answer into our original equations or ones we developed along the way. Using the *v-i* relation for the output resistor, we obtain the quantity we seek.

$$\boxed{v_{\text{out}} = \frac{R_2}{R_1 + R_2} v_{\text{in}}}$$

Question 3.2 {**248**} *Referring back to Figure 3.3 {27}, a circuit should serve some useful purpose. What kind of system does our circuit realize and, in terms of element values, what are the system's parameter(s)?*

This result, and the values of other currents and voltages in this circuit as well, have profound implications.

- Resistors connected in such a way that current from one must flow *only* into another—currents in all resistors connected this way have the same magnitude—are said to be connected in *series*. For two series-connected resistors in the example, *the voltage across one resistor equals the ratio of that resistor's value and the sum of resistances times the voltage across the series combination.* Although we derived this result when a voltage source is placed across the series combination, *it applies in general.* This concept is so pervasive it has a name: *voltage divider.*

---

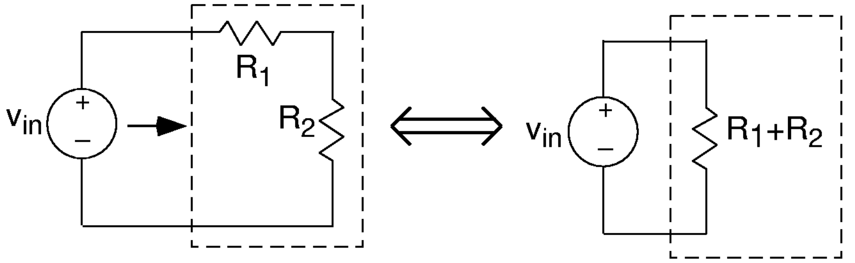
[*] Yes, we temporarily eliminate the quantity we seek. Though not obvious, it is the simplest way to solve the equations.

- The *input-output relationship* for this system, found in this particular case by voltage divider, takes the form of a ratio of the output voltage to the input voltage.

  $$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{R_2}{R_1 + R_2}$$

  In this way, we express how the components used to build the system affect the input-output relationship. Because this analysis was made with ideal circuit elements, we might expect this relation to break down if the input amplitude is too high (Will the circuit survive if the input changes from 1 volt to one million volts?) or if the source's frequency becomes too high. In any case, this important way of expressing input-output relationships —as a ratio of output to input —pervades circuit and system theory.

- The current i_1 is the current flowing out of the voltage source. Because it equals i_2, we have that $v_{\text{in}}/i_1 = R_1 + R_2$: *The series combination of two resistors acts, as far as the voltage source is concerned, as a* ***single*** *resistor having a value equal to the sum of the two resistances.* This result is the first of several *equivalent circuit ideas*: In many cases, a complicated circuit when viewed from its terminals (the two places to which you might attach a source) appears to be a single circuit element (at best) or a simple combination of elements at worst. Thus, the equivalent circuit for a series combination of resistors is a single resistor having a resistance equal to the sum of its component resistances. Thus, the circuit the voltage source "feels" (through the current drawn from it) is a single resistor having resistance $R_1 + R_2$. Note that in making this equivalent circuit, the output voltage can no longer be defined: The output resistor labeled R_2 no longer appears. Thus, this equivalence is made strictly from the voltage source's viewpoint.



A second interesting simple circuit (Figure 3.4 {31}) has two resistors connected side-by-side, what we will term a *parallel* connection, rather than in series.

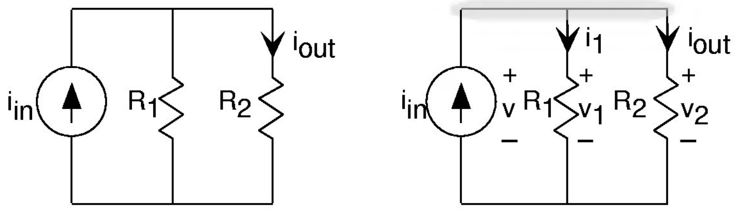


Figure 3.4 A simple parallel circuit.

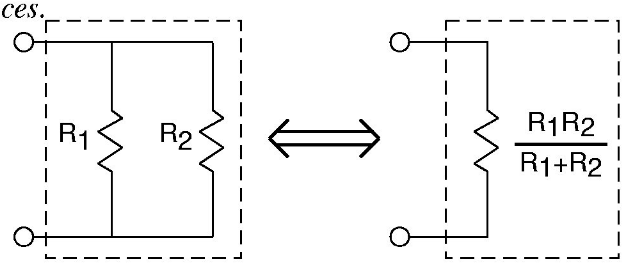
Here, applying KVL rveals that all the voltages are identical: $v_1 = v$ and $v_2 = v$. This result typifies parallel connections. To write the KCL equation, note that the top node consists of the entire upper interconnection section. The KCL equation is $i_{\text{in}} - i_1 - i_2 = 0$. Using the *v*-*i* relations, we find that

$$\boxed{i_{\text{out}} = \frac{R_1}{R_1 + R_2} i_{\text{in}} .}$$

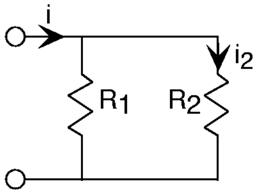
Question 3.3 {**248**} *Suppose that you replaced the current source in Figure 3.4 {31} by a voltage source. How would* i_{out} *be related to the source voltage? Based on this result, what purpose does this revised circuit have?*

This circuit highlights some important properties of parallel circuits.

- You can easily show that the parallel combination of R_1 and R_2 has the *v*-*i* relation of a resistor having resistance $(1/R_1 + 1/R_2)^{-1} = R_1 R_2/(R_1 + R_2)$. A shorthand notation for this quantity is $R_1 \| R_2$. As the reciprocal of resistance is conductance (recall the definitions {24}), we can say that *for a parallel combination of resistors, the equivalent conductance is the sum of the conductances.*



- Similar to voltage divider for series resistances, we have *current divider* for parallel resistances. The current through a resistor in parallel with another is the ratio of the conductance of the first to the sum of the conductances. Thus, for the depicted circuit, $i_2 = G_2/(G_1 + G_2)i$. Expressed in terms of resistances, current divider takes the form of the resistance of the *other* resistor divide by the sum of resistances: $i_2 = R_1/(R_1 + R_2)i$.

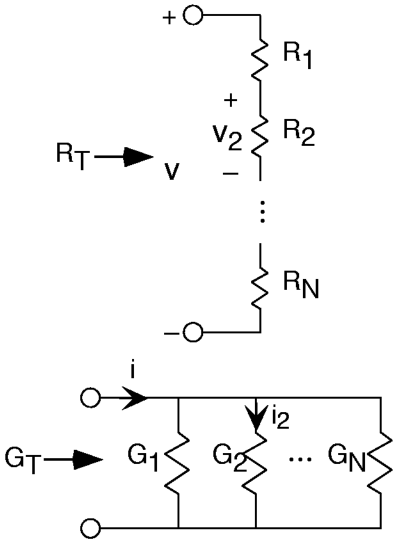


To summarize, the series and parallel combination results shown in Figure 3.5 are easy to remember and very useful. Keep in mind that for series combinations, voltage and resistance are the key quantities, while for parallel combinations current and conductance are more important. In series combinations, the currents through each element are the same; in parallel ones, the voltages are the same.

Question 3.4 {**248**} *Contrast a series combination of resistors with a parallel one. Which variable (voltage or current) is the same for each and which differs? What are the equivalent resistances? When resistors are placed in series, is the equivalent resistance bigger, in between, or smaller than the component resistances? What is this relationship for a parallel combination?*

Let's now suppose we want to pass the output signal into a voltage measurement device, such as an oscilloscope or a voltmeter. In system-theory terms, we want to

---



$$R_T = \sum_{n=1}^{N} R_n \qquad v_n = \frac{R_n}{R_T} v$$

$$G_T = \sum_{n=1}^{N} G_n \qquad i_n = \frac{G_n}{G_T} i$$

Figure 3.5 Series and parallel combination rules.

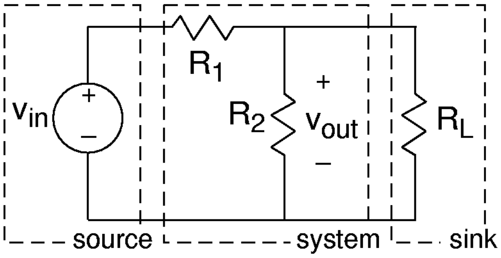


Figure 3.6 The simple attenuator circuit shown in Figure 3.3 {27} is attached to an oscilloscope's input.

pass our circuit's output to a sink. For most applications, we can represent these measurement devices as a resistor, with the current passing through it driving the measurement device through some type of display. In circuits, a sink is called a *load*; thus, we describe a system-theoretic sink as a load resistance R_L. Thus, we have a complete system built from a cascade of three systems: a source, a signal processing system (simple as it is), and a sink.

We must analyze afresh how this revised circuit, shown in Figure 3.6 {33}, works. Rather than defining eight variables and solving for the current in the load resistor, let's take a hint from our previous analysis. Resistors R_2 and R_L are in a *parallel* configuration: The voltages across each resistor are the same while the currents are not. Because the voltages are the same, we can find the current through each from their *v-i* relations: $i_2 = v_{\text{out}}/R_2$ and $i_L = v_{\text{out}}/R_L$. Considering the node where all three resistors join, KCL says that the sum of the three currents must equal zero. Said another way, the current entering the node through R_1 must equal the sum of the other two currents leaving the node. Therefore, $i_1 = i_2 + i_L$, which means that $i_1 = v_{\text{out}} \cdot (1/R_2 + 1/R_L)$.

Let R_{eq} denote the equivalent resistance of the parallel combination of R_2 and R_L. Using R_1's *v-i* relation, the voltage across it is $v_1 = R_1 v_{\text{out}}/R_{\text{eq}}$. The KVL equation written around the leftmost loop has $v_{\text{in}} = v_1 + v_{\text{out}}$; substituting for v_1, we find

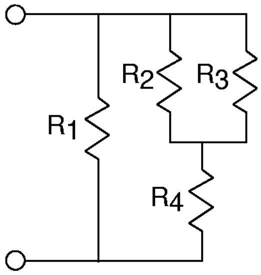
$$v_{\text{in}} = v_{\text{out}} \cdot \left(\frac{R_1}{R_{\text{eq}}} + 1\right)$$

$$\text{or} \quad \frac{v_{\text{out}}}{v_{\text{in}}} = \frac{R_{\text{eq}}}{R_1 + R_{\text{eq}}}$$

Again, we have the input-output relationship for our entire system having the form of voltage divider, but it does *not* equal the input-output relation that we had before we incorporated the voltage measurement device. We can not measure voltages reliably unless the measurement device has little effect on what we are trying the measure. We should look more carefully to determine if any values for the load resistance would lessen its impact on the circuit. Comparing the input-output relations before and after, what we need is $R_{\text{eq}} \approx R_2$. As $R_{\text{eq}} = (1/R_2 + 1/R_L)^{-1}$, the approximation would apply if $1/R_2 \gg 1/R_L$ or $R_2 \ll R_L$. This is the condition we seek: *Voltage measurement devices must have large resistances compared with that of the resistor across which the voltage is to be measured.*

Question 3.5 {**248**} *Let's be more precise: How much larger would a load resistance need to be to affect the input-output relation by less than 10%? by less than 1%?*

Example



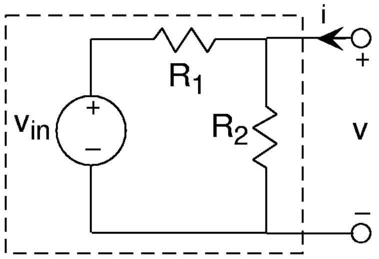
We want to find the total resistance of the example circuit. To apply the series and parallel combination rules, it is best to first determine the circuit's structure: What is in series with what and what is in parallel with what at both small- and large-scale views. We have R_2 in parallel with R_3; this combination is in series with R_4. This series combination is in parallel with R_1. Note that in determining this structure, we started *away* from the terminals, and worked toward them. In most cases, this approach works well; try it first. The total resistance expression mimics the structure:

$$\begin{aligned} R_T &= R_1 \| (R_2 \| R_3 + R_4) \\ &= \frac{R_1 R_2 R_3 + R_1 R_2 R_4 + R_1 R_3 R_4}{R_1 R_2 + R_1 R_3 + R_2 R_3 + R_2 R_4 + R_3 R_4} \end{aligned}$$

Such complicated expressions typify circuit "simplifications." A simple check for accuracy is the units: Each component of the numerator should have the same units (here Ω^3) as well as in the denominator (Ω^2). The entire expression is to have units of resistance; thus, the ratio of the numerator's and denominator's units should be ohms. Checking units does not guarantee accuracy, but can catch many errors.

Another valuable lesson emerges from this example concerning the difference between cascading systems and cascading circuits. In system theory, systems can be cascaded without changing the input-output relation of intermediate systems. In cascading circuits, this ideal is rarely true *unless* the circuits are so *designed*. Design is in the hands of the engineer; he or she must recognize what have come to be known as loading effects. In our simple circuit, you might think that making the resistance R_L large enough would do the trick. Because the resistors R_1 and R_2 can have virtually any value, you can never make the resistance of your voltage measurement device big enough. Said another way, *a circuit cannot be designed in isolation that will work in cascade with all other circuits*. Electrical engineers deal with this situation through the notion of *specifications*: Under what conditions will the circuit perform as designed? Thus, you will find that oscilloscopes and voltmeters have their internal resistances clearly stated, enabling you to determine whether the voltage you measure closely equals what was present before they were attached to your circuit. Furthermore, since our resistor circuit functions as an attenuator, with the attenuation (a fancy word for gains less than one) depending only on the ratio of the two resistor values $\left(R_2/(R_1+R_2) = (1+R_1/R_2)^{-1}\right)$, we can select *any* values for the two resistances we want to achieve the desired attenuation. The designer of this circuit must thus specify not only what the attenuation is, but also the resistance values employed so that integrators —people who put systems together from component systems —can combine systems together and have a chance of the combination working.

3.6 Equivalent Circuits



We have found that the way to think about circuits is to locate and group parallel and series resistor combinations. Those resistors not involved with variables of interest can be collapsed into a single resistance. This result is known as an *equivalent circuit*: from the viewpoint of a pair of terminals, a group of resistors functions as a single resistor, the resistance of which can usually be found by applying the parallel and series rules.

This result generalizes to include sources in a very interesting and useful way. Let's consider again our attenuator circuit, this time from the viewpoint of the output terminals. We want to find the *v*-*i* relation for the output terminal pair, and then find the equivalent circuit for the boxed circuit. To perform this calculation, use the circuit laws and element relations, but do not assume that the output terminals have nothing connected to them. The schematic may seem to suggest that nothing is attached to the output terminals; when you see the current i defined as it is, the presumption is that a circuit is about to be attached. Therefore, we seek the relation between v and i that describes the kind of element that lurks within the dashed box. The result is

$$v = (R_1 \| R_2)i + \frac{R_2}{R_1 + R_2} v_{\text{in}} \ .$$

If the source were zero, it could be replaced by a short circuit, which would confirm that the circuit does indeed function as a parallel combination of resistors. However, the source's presence means that the circuit is *not* well modeled as a resistor.

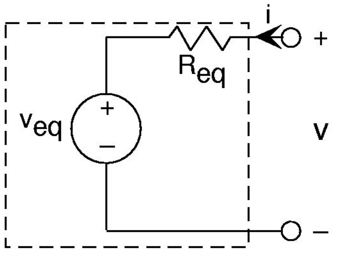


Figure 3.7 The Thévenin equivalent circuit.

If we consider the simple circuit of Figure 3.7, we find it has the *v*-*i* relation at its terminals of

$$\boxed{v = R_{\text{eq}} i + v_{\text{eq}}}$$

Comparing the two *v*-*i* relations, we find that they have the same form. In this case the *Thévenin equivalent resistance* is $R_{\text{eq}} = R_1 \| R_2$ and the *Thévenin equivalent source* has voltage $v_{\text{eq}} = \frac{R_2}{R_1+R_2} v_{\text{in}}$. Thus, from the terminals, you cannot distinguish the two circuits. Because the equivalent circuit has fewer elements, it is easier to analyze and understand than any other alternative.

For *any* circuit containing resistors and sources, the *v*-*i* relation will be of this form and the *Thévenin equivalent circuit* for any such circuit is that of Figure 3.7. This equivalence applies no matter how many sources or resistors may be present in the circuit. In the current example, we know the circuit's construction and element values, and derived the equivalent source and resistance. Because Thévenin's theorem applies in general, we should be able to make measurements or calculations *only from the terminals* to determine the equivalent circuit.

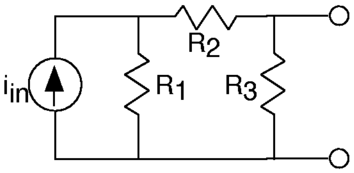
To be more specific, consider the equivalent circuit of Figure 3.7. Let the terminals be open-circuited, which has the effect of setting the current i to zero. Because no current flows through the resistor, the voltage across it is zero (remember, Ohm's Law says that $v = Ri$). Consequently, by applying KVL we have that the so-called open-circuit voltage v_{oc} equals the Thévenin equivalent voltage. Now consider the situation when we set the terminal voltage to zero (short-circuit it) and measure the resulting current. Referring to the equivalent circuit, the source voltage now appears entirely across the resistor, leaving the short-circuit current to be

$i_{sc} = -v_{eq}/R_{eq}$. From this property, we can determine the equivalent resistance.

$$v_{eq} = v_{oc}$$
$$R_{eq} = -\frac{v_{oc}}{i_{sc}}$$

Question 3.6 {**248**} *Use the open/short-circuit approach to derive the Thévenin equivalent of the circuit shown on page {35}.*

Example



For the depicted circuit, let's derive its Thévenin equivalent two different ways. Starting with the open/short-circuit approach, let's first find the open-circuit voltage v_{oc}. We have a current divider relationship as R_1 is in parallel with the series combination of R_2 and R_3. Thus, $v_{oc} = i_{in} \cdot R_3 R_1/(R_1 + R_2 + R_3)$. When we short-circuit the terminals, no voltage appears across R_3, and thus no current flows through it. In short, R_3 does not affect the short-circuit current, and can be eliminated. We again have a current divider relationship: $i_{sc} = -i_{in} R_1/(R_1 + R_2)$. Thus, the Thévenin equivalent resistance is $R_3(R_1 + R_2)/(R_1 + R_2 + R_3)$.

To verify, let's find the equivalent resistance by reaching inside the circuit and setting the current source to zero. Because the current is now zero, we can replace the current source by an open circuit. From the viewpoint of the terminals, resistor R_3 is now in parallel with the series combination of R_1 and R_2. Thus, $R_{eq} = R_3 \| (R_1 + R_2)$, and we obtain the same result.

As you might expect, equivalent circuits come in two forms: the voltage-source oriented Thévenin equivalent and the current-source oriented *Norton equivalent* (Figure 3.8 {38}). To derive the latter, the *v-i* relation for the Thévenin equivalent can be written as

$$v = R_{eq} i + v_{eq} \quad \text{or} \quad i = \frac{v}{R_{eq}} - i_{eq} \, ,$$

where $i_{eq} = \frac{v_{eq}}{R_{eq}}$ is the Norton equivalent source. The Norton equivalent shown in Figure 3.8 {38} can be easily shown to have this *v-i* relation. Note that both variations have the same equivalent resistance. The short-circuit current equals the negative of the Norton equivalent source.

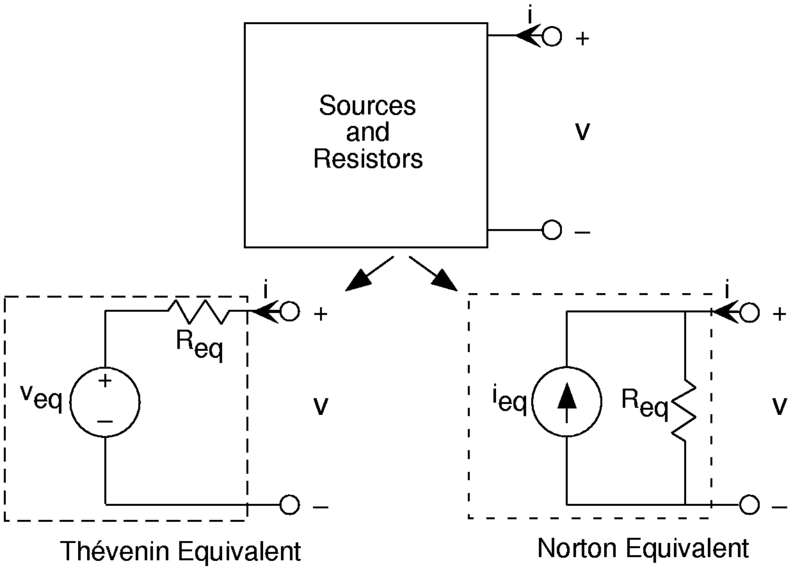


Figure 3.8 *All* circuits containing sources and resistors can be described by simpler equivalent circuits. Choosing the one to use depends on the application, not on what is actually inside the circuit.

Question 3.7 {248} *Find the Norton equivalent circuit for the example on page 37.*

Equivalent circuits can be used in two basic ways. The first is to simplify the analysis of a complicated circuit by realizing the *any* portion of a circuit can be described by either a Thévenin or Norton equivalent. Which one is used depends on whether what is attached to the terminals is a series configuration (making the Thévenin equivalent the best) or a parallel one (making Norton the best).

Another application is modeling. When we buy a flashlight battery, either equivalent circuit can accurately describe it. These models help us understand the limitations of a battery. Since batteries are labeled with a voltage specification, they should serve as voltage sources and the Thévenin equivalent serves as the natural choice. If a load resistance R_L is placed across its terminals, the voltage output can be found using voltage divider: $v = v_{\text{eq}} R_L / (R_L + R_{\text{eq}})$. If we have a load resistance much larger than the battery's equivalent resistance, then, to a good approximation, the battery does serve as a voltage source. If the load resistance is much smaller, we certainly don't have a voltage source (the output voltage depends directly on the load resistance). Consider now the Norton equivalent; the current through the load resistance is given by current divider, and equals $i = -i_{\text{eq}} R_{\text{eq}} / (R_L + R_{\text{eq}})$. For a current that does not vary with the load resistance, this resistance should be much smaller than the equivalent resistance. If the load resistance is comparable to the equivalent resistance, the battery serves *neither* as a voltage source or a current course. Thus, when you buy a battery, you get a voltage source if its equivalent resistance is much *smaller* than the equivalent resistance of the circuit you attach it to. On the other hand, if you attach it to a circuit having a small equivalent resistance, you bought a current source.

3.7 Circuits with Capacitors and Inductors

Let's consider a circuit having something other than resistors and sources. Because of KVL, we know that $v_{\text{in}} = v_R + v_{\text{out}}$. The current through the capacitor is given by $i = C\frac{dv_{\text{out}}}{dt}$, and this current equals that passing through the resistor. Substituting $v_R = Ri$ into the KVL equation and using the *v-i* relation for the capacitor, we arrive at

$$RC\frac{dv_{\text{out}}}{dt} + v_{\text{out}} = v_{\text{in}} \ .$$

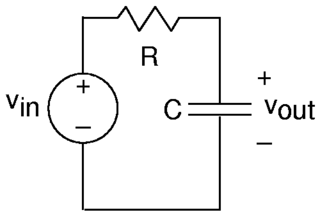


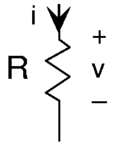
Figure 3.9 A simple RC circuit.

The input-output relation for circuits involving energy storage elements takes the form of an ordinary differential equation, which we must solve to determine what the output voltage is for a given input. In contrast to resistive circuits, where we obtain an *explicit* input-output relation, we now have an *implicit* relation that requires more work to obtain answers.

At this point, we could learn how to solve differential equations. Note first that even finding the differential equation relating an output variable to a source is often very tedious. The parallel and series combination rules that apply to resistors don't directly apply when capacitors and inductors occur. We would have to slog our way through the circuit equations, simplifying them until we finally found the equation that related the source(s) to the output. What was discovered at the turn of the twentieth century was a method for not only easily finding the differential equation, but also simplified the solution process in the most common situation. Although not original with him, Charles Steinmetz presented the key paper describing the *impedance* approach in 1893. The impedance concept is very important, extending its reach far beyond circuits.

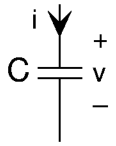
3.7.1 Impedance

The idea is to pretend that all sources in the circuit are complex exponentials having the *same* frequency. Thus for our example RC circuit (Figure 3.9 {39}), let



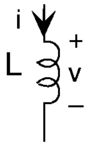
Resistor

$Z_R = R$



Capacitor

$Z_C = \frac{1}{j2\pi fC}$



Inductor

$Z_L = j2\pi fL$

$v_{\text{in}} = V_{\text{in}}e^{j2\pi ft}$. The complex amplitude V_{in} determines the size of the source and its phase. The critical consequence of assuming that sources have this form is that *all* voltages and currents in the circuit are also complex exponentials, having amplitudes governed by KVL, KCL, and the *v*-*i* relations and the same frequency as the source. To appreciate why this should be true, let's investigate how each circuit element behaves when either the voltage or current is a complex exponential. For the resistor, $v = Ri$. Let $v = Ve^{j2\pi ft}$; then $i = \frac{V}{R}e^{j2\pi ft}$. As stated, if the resistor's voltage is a complex exponential, so is the current, with an amplitude $I = \frac{V}{R}$ (determined by the resistor's *v*-*i* relation) and the same frequency as the voltage. Clearly, if the current were assumed to be a complex exponential, so would the voltage. For a capacitor, $i = C\frac{dv}{dt}$. Letting the voltage be a complex exponential, we have $i = CVj2\pi fe^{j2\pi ft}$. The amplitude of this complex exponential is $I = CVj2\pi f$. Finally, for the inductor, where $v = L\frac{di}{dt}$, assuming the current to be a complex exponential results in the voltage having the form $v = LIj2\pi fe^{j2\pi ft}$, making its complex amplitude $V = LIj2\pi f$.

The major consequence of assuming complex exponential voltage and currents is that the ratio $Z = V/I$ for each element does not depend on time. This quantity is known as the element's *impedance*. The impedance is, in general, a complex-valued, frequency-dependent quantity. For example, the magnitude of the capacitor's impedance is inversely related to frequency, and has a phase of $-\pi/2$. This observation means that if the current is a complex exponential and has constant amplitude, the amplitude of the voltage decreases with frequency.

Let's consider Kirchoff's circuit laws. When voltages around a loop are all complex exponentials of the same frequency, we have

$$\sum_n v_n = \sum_n V_n e^{j2\pi ft} = 0$$

$$\text{which means } \sum_n V_n = 0 \;;$$

the complex amplitudes of the voltages obey KVL. We can easily imagine that the complex amplitudes of the currents obey KCL.

What we have discovered is that source(s) equaling a complex exponential of the same frequency forces all circuit variables to be complex exponentials of the same frequency. Consequently, the ratio of voltage to current for each element equals the ratio of their complex amplitudes, which depends only on the source's frequency and element values. This situation occurs because the circuit elements are linear and time-invariant. For example, suppose we had a circuit element where the voltage equaled the square of the current: $v(t) = Ki^2(t)$. If $i(t) = Ie^{j2\pi ft}$, $v(t) = KI^2e^{j2\pi 2ft}$, meaning that voltage and current no longer had the same frequency and that their ratio was time-dependent.

> **Because for linear circuit elements the complex amplitude of voltage is proportional to the complex amplitude of current—$V = ZI$—circuit elements behave as if they were resistors, where instead of resistance, we use impedance. Because complex amplitudes for voltage and current also obey Kirchoff's laws, we can solve circuits using voltage and current divider and the series and parallel combination rules by considering the elements to be impedances.**

As we unfold the impedance story, we'll see that this powerful result of Steinmetz's greatly simplifies solving circuits, alleviates us from solving differential equations, and suggests a general way of thinking about circuits. Because of the importance of this approach, let's go over how it works.

- Even though it's not, pretend the source is a complex exponential. We do this because the impedance approach simplifies finding how input and output are related. If it were a voltage source having voltage $v_{\text{in}} = \mathrm{p}(t)$ (a pulse), still let $v_{\text{in}} = V_{\text{in}}e^{j2\pi ft}$. We'll learn how to "get the pulse back" later.
- With a source equaling a complex exponential, *all* variables in a linear circuit will also be complex exponentials having the *same* frequency. The circuit's only remaining "mystery" is what each variable's complex amplitude might be. To find these, we consider the source to be a complex number (V_{in} here) and the elements to be impedances.
- We can now solve using series and parallel combination rules how the complex amplitude of any variable relates to the sources complex amplitude.

A common error in using impedances is to keep the time-dependent part, the complex exponential, in the fray. The entire point of using impedances is to get rid of them *in writing circuit equations and in the subsequent algebra.* The complex exponentials are there *implicitly* (they're behind the scenes). Only after we find the result do we raise the curtain and put things back to together again. In short, *when solving circuits using impedances, t should not appear except for the beginning and end.*

To illustrate the impedance approach, we return to our RC circuit (Figure 3.9 $\{39\}$), we assume that $v_{\text{in}} = V_{\text{in}} e^{j2\pi ft}$. Using impedances, the complex amplitude of the output voltage V_{out} can be found using voltage divider:

$$\begin{aligned} V_{\text{out}} &= \frac{Z_C}{Z_C + Z_R} V_{\text{in}} \\ &= \frac{\frac{1}{j2\pi fC}}{\frac{1}{j2\pi fC} + R} V_{\text{in}} \\ &= \frac{1}{j2\pi fRC + 1} V_{\text{in}} \end{aligned}$$

If we return to the differential equation we found for this circuit $\{39\}$, letting the output and input voltages be complex exponentials, we obtain the same relationship between their complex amplitudes. Thus, using impedances is equivalent to using the differential equation and solving it when the source is a complex exponential.

In fact, we can find the differential equation *directly* using impedances. If we cross-multiply the relation between input and output amplitudes,

$$V_{\text{out}} \left(j2\pi fRC + 1 \right) = V_{\text{in}} ,$$

and then put the complex exponentials back in, we have

$$RC \left(j2\pi f V_{\text{out}} e^{j2\pi ft} \right) + V_{\text{out}} e^{j2\pi ft} = V_{\text{in}} e^{j2\pi ft} .$$

In the process of defining impedances, note that the factor $j2\pi f$ arises from the *derivative* of a complex exponential. We can reverse the impedance process, and revert back to the differential equation.

$$RC \frac{dv_{\text{out}}}{dt} + v_{\text{out}} = v_{\text{in}}$$

This is the same equation we derived much more tediously at the beginning of this section. Finding the differential equation relating output to input is far simpler when we use impedances than with any other technique.

Question 3.8 $\{$**248**$\}$ *Suppose you had an expression where a complex amplitude was* divided *by $j2\pi f$. How did this division arise?*

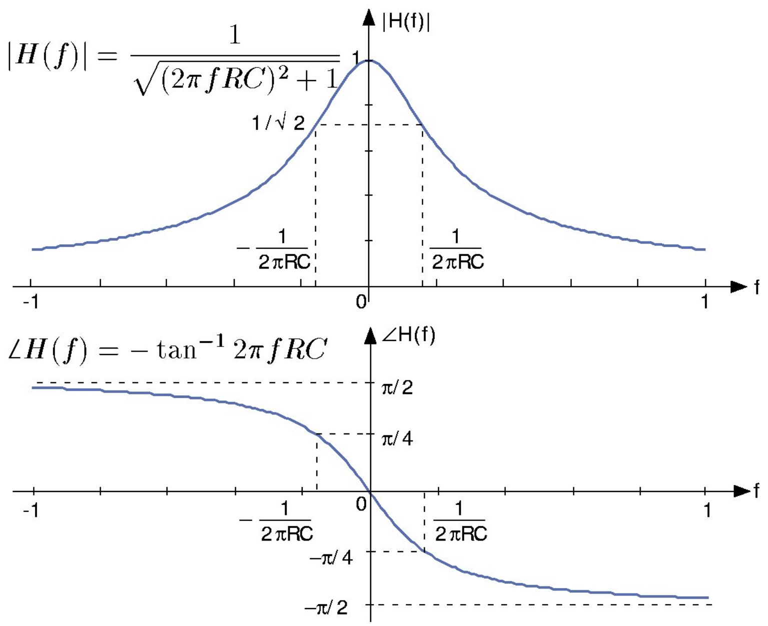


Figure 3.10 Magnitude and phase of the transfer function of the RC circuit shown in Figure 3.9 {39} when $RC = 1$.

3.7.2 Transfer Function

The ratio of the output and input amplitudes, known as the *transfer function* or the *frequency response*, is given by

$$\boxed{\frac{V_{\text{out}}}{V_{\text{in}}} = H(f) = \frac{1}{j2\pi fRC + 1}} \tag{3.1}$$

Implicit in using the transfer function is that the input is a complex exponential, thereby resulting in an output having the same frequency. The transfer function reveals how the circuit modifies the input amplitude in creating the output. Thus, the transfer function *completely* describes how the circuit processes the input complex exponential to produce the output complex exponential. The circuit's function is thus summarized by the transfer function. In fact, circuits are often designed to meet transfer function specifications. Because transfer functions are complex-valued, frequency-dependent quantities, we can better appreciate a circuit's function by examining the magnitude and phase of its transfer function (Figure 3.10). Several things to note about this transfer function.

- We can compute the frequency response for both positive and negative frequencies. Recall that sinusoids consist of the sum of two complex exponentials, one having the negative frequency of the other. We will consider how the circuit acts on a sinusoid soon. Do note that the magnitude has *even symmetry*: The negative frequency portion is a mirror image of the positive frequency portion: $|H(-f)| = |H(f)|$. This property of this specific example occurs for all transfer functions associated with circuits. Consequently, we don't need to plot the negative frequency component.
- The magnitude equals $1/\sqrt{2}$ when $2\pi fRC = 1$ (the two terms in the denominator of the magnitude are equal). The frequency $f_c = 1/2\pi RC$ defines the boundary between two operating ranges.
  - For frequencies below this frequency, the circuit does not much alter the amplitude of the complex exponential source.
  - For frequencies greater than f_c, the circuit strongly attenuates the amplitude. Thus, when the source frequency is in this range, the circuit's output has a much smaller amplitude than that of the source.

  For these reasons, this frequency is known as the *cutoff frequency*.
- The cutoff frequency depends *only* on the product of the resistance and the capacitance. Thus, a cutoff frequency of 1 kHz occurs when $1/2\pi RC = 10^3$ or $RC = 10^{-3}/2\pi = 1.59 \times 10^{-4}$. Thus resistance-capacitance combinations of 1.59 kΩ and 100 nF and 100 Ω and 1.59 μF result in the same cutoff frequency.
- The phase shift caused by the circuit at the cutoff frequency precisely equals $-\pi/4$. Thus, below the cutoff frequency, phase is little affected, but at higher frequencies, the phase shift caused by the circuit becomes $-\pi/2$. This phase shift corresponds to the difference between a cosine and a sine.
- If the source consists of two (or more) signals, we know from linear system theory that the output voltage equals the sum of the outputs produced by each signal alone. In short, linear circuits are a special case of linear systems, and therefore superposition applies. In particular, suppose these component signals are complex exponentials, each of which has a frequency different from the others. The transfer function portrays how the circuit affects the amplitude and phase of each component, allowing us to understand how the circuit works on a complicated signal. Those components having a frequency less than the cutoff frequency pass through the circuit with little modification while those having higher frequencies are suppressed. The circuit is said to act as a *filter*, filtering the source signal based on the frequency of each

component complex exponential. Because low frequencies pass through the filter, we call it a *lowpass* filter to express more precisely its function.

One very important special case occurs when the input voltage is a sinusoid. Because a sinusoid is the sum of two complex exponentials, each having a frequency equal to the negative of the other, we can directly predict from examining the transfer function the output voltage. If the source is a sine wave, we know that

$$v_{\text{in}}(t) = A\sin 2\pi ft = \frac{A}{2j}\left(e^{j2\pi ft} - e^{-j2\pi ft}\right) .$$

Since the input is the sum of two complex exponentials, we know that the output is also a sum of two similar complex exponentials, the only difference being that the complex amplitude of each is multiplied by the transfer function evaluated at each exponential's frequency.

$$v_{\text{out}}(t) = \frac{A}{2j}H(f)e^{j2\pi ft} - \frac{A}{2j}H(-f)e^{-j2\pi ft}$$

As noted earlier, the transfer function is most conveniently expressed in polar form: $H(f) = |H(f)|e^{j\angle H(f)}$. Furthermore, $|H(-f)| = |H(f)|$ (even symmetry of the magnitude) and $\angle H(-f) = -\angle H(f)$ (odd symmetry of the phase). The output voltage expression simplifies to

$$\begin{aligned} v_{\text{out}}(t) &= \frac{A}{2j}|H(f)|e^{j2\pi ft+\angle H(f)} - \frac{A}{2j}|H(f)|e^{-j2\pi ft-\angle H(f)} \\ &= A|H(f)|\sin\bigl(2\pi ft + \angle H(f)\bigr) \end{aligned} \tag{3.2}$$

The circuit's output to a sinusoidal input is also a sinusoid, having an gain equal to the magnitude of the circuit's transfer function evaluated at the source frequency and a phase equal to the phase of the transfer function at the source frequency. It will turn out that this input-output relation description applies to any linear circuit having a sinusoidal source.

Question 3.9 {**248**} *This input-output property is a special case of a more general result. Show that if the source can be written as the imaginary part of a complex exponential—*$v_{in}(t) = \mathit{Im}\left[Ve^{j2\pi ft}\right]$*—the output is given by* $v_{out}(t) = \mathit{Im}\left[VH(f)e^{j2\pi ft}\right]$*. Show that a similar result also holds for the real part.*

3.7.3 Equivalent Circuits Revisited

When we have circuits with capacitors and/or inductors as well as resistors and sources, Thévenin and Norton equivalent circuits can still be defined in using

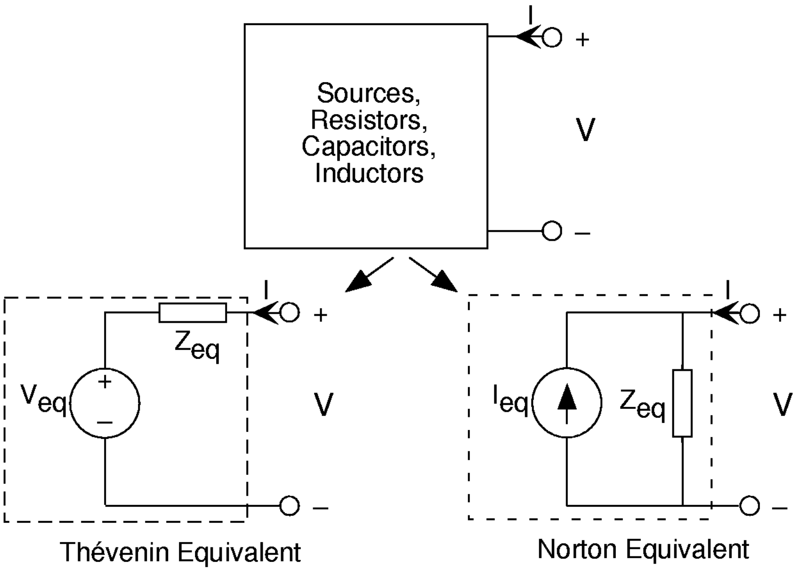

Sources,
Resistors,
Capacitors,
Inductors
I
+
V
–
Veq
Zeq
Thévenin Equivalent
Ieq
Zeq
Norton Equivalent

Figure 3.11 Comparing this figure with Figure 3.8 {38}, we see two differences. First of all, more circuits (all those containing linear elements in fact) have equivalent circuits that contain equivalents. Secondly, the terminal and source variables are now complex amplitudes, which carries the implicit assumption that the voltages and currents are single complex exponentials, all having the same frequency.

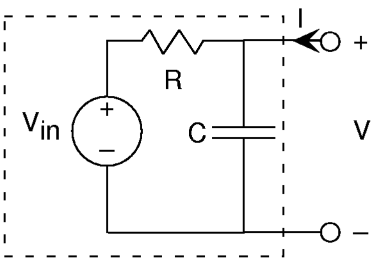
---

impedances and complex amplitudes for voltage and currents. For any circuit containing sources, resistors, capacitors, and inductors, the input-output relation for the *complex amplitudes* of the terminal voltage and current is

$$V = Z_{\text{eq}} I + V_{\text{eq}}$$
$$I = \frac{V}{Z_{\text{eq}}} - I_{\text{eq}}$$

with $V_{\text{eq}} = Z_{\text{eq}} I_{\text{eq}}$. Thus, we have Thévenin and Norton equivalent circuits as shown in Figure 3.11.

Example



For the same circuit we have been analyzing, let's find its Thévenin and Norton equivalent circuits. The open-circuit voltage and short-circuit current techniques still work, except here we use impedances and complex amplitudes. The open-circuit voltage corresponds to the transfer function we have already found. When we short the terminals, the capacitor no longer has any effect on the circuit, and the short-circuit current I_{sc}equals $-V_{\text{out}}/R$. The equivalent impedance can be found by setting the source to zero, and finding the impedance using series and parallel combination rules. In our case, the resistor and capacitor are in parallel once the voltage source is removed (setting it to zero amounts to replacing it with a short-circuit). Thus, $Z_{\text{eq}} = R||(1/j2\pi fC) = R/(1+j2\pi fRC)$. Consequently, we have

$$V_{\text{eq}} = \frac{1}{1+j2\pi fRC}V_{\text{in}} \quad I_{\text{eq}} = \frac{1}{R}V_{\text{in}} \quad Z_{\text{eq}} = \frac{R}{1+j2\pi fRC}$$

Again, we should check the units of our answer. Note in particular that $j2\pi fRC$ must be dimensionless; is it?

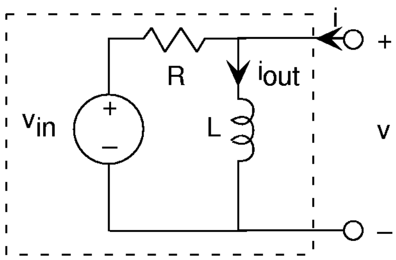
3.7.4 Summary

The notion of impedance arises when we assume the sources are complex exponentials. This assumption may seem restrictive; what would we do if the source were a unit step? When we use impedances to find the transfer function between the source and the output variable, we can derive from it the differential equation that relates input and output. The differential equation applies no matter what the source may be. As we have argued, it is far simpler to use impedances to find the differential equation (because we can use series and parallel combination rules) that any other method. In this sense, we have not lost anything by temporarily pretending the source is a complex exponential.

Subsequent sections reveal the surprising result that we can also solve the differential equation using impedances! Thus, despite the apparent restrictiveness of impedances, assuming complex exponential sources actually is quite general.

We have also found the ease of calculating the output for sinusoidal inputs through the use of the transfer function. Once we find the transfer function, we can write the output directly as indicated by (3.2) $\{45\}$.

Example



Let's apply these results to a final example, in which the input is a voltage source and the output is the inductor current. The source voltage equals $v_{\text{in}} = 2\cos(2\pi 60t) + 3$. We want the circuit to pass constant (offset) voltage essentially unaltered (save for the fact the output is a current rather than a voltage) and remove the 60 Hz term. Because the input is the sum of *two* sinusoids—a constant is a zero-frequency cosine—our approach is

1. find the transfer function using impedances;
2. use it to find the output due to each input component;
3. add the results;
4. find element values that accomplish our design criteria.

Because the circuit is a series combination of elements, let's use voltage divider to find the transfer function between V_{in} and V, then use the *v*-*i* relation of the inductor to find its current.

$$\begin{aligned}\frac{I_{\text{out}}}{V_{\text{in}}} &= \underbrace{\frac{j2\pi fL}{R + j2\pi fL}}_{\text{voltage divider}} \cdot \underbrace{\frac{1}{j2\pi fL}}_{\text{inductor admittance}} \\ &= \underbrace{\frac{1}{j2\pi fL + R}}_{H(f)}\end{aligned}$$

[Do the units check?] The form of this transfer function should be familiar; it is a lowpass filter, and it will perform our desired function once we choose element values properly.

The constant term is easiest to handle. The output is given by $3 \cdot H(0) = 3/R$. Thus, the value we choose for the resistance will determine the scaling factor of how voltage is converted into current. For the 60 Hz component signal, the output current is $2|H(60)|\cos\bigl(2\pi 60t + \angle H(60)\bigr)$. The total output due to our source is

$$i_{\text{out}} = 2|H(60)|\cos\bigl(2\pi 60t + \angle H(60)\bigr) + 3H(0)\ .$$

The cutoff frequency for this filter occurs when the real and imaginary parts of the transfer function's denominator equal each other. Thus, $2\pi f_c L = R$, which gives $f_c = R/2\pi L$. We want this cutoff frequency to be much less than 60 Hz. Suppose we place it at, say, 10 Hz. This specification would require the component values to be related by $R/L = 20\pi = 62.8$. The transfer function at 60 Hz would be

$$\begin{aligned}\left|\frac{1}{j2\pi 60L + R}\right| &= \frac{1}{R}\cdot\left|\frac{1}{6j+1}\right| \\ &= \frac{1}{R}\cdot\frac{1}{\sqrt{37}} \\ &\approx 0.16\frac{1}{R},\end{aligned}$$

which yield an attenuation (relative to the gain at zero frequency) of about $1/6$, and result in an output amplitude of $0.3/R$ relative to the constant term's amplitude of $3/R$. A factor of 10 relative size between the two components seems reasonable. Having a 100 mH inductor would require a 6.28 Ω resistor. An easily available resistor value is 6.8 Ω; thus, this choice results in cheaply and easily purchased parts. To make the resistance bigger would require a proportionally larger inductor. Unfortunately, even a 1 H inductor is physically large; consequently low cutoff frequencies require small-valued resistors and large-valued inductors. The choice made here represents only one compromise.

The phase of the 60 Hz component will very nearly be $-\pi/2$, leaving it to be $(0.3/R)\cos(2\pi 60t - \pi/2) = (0.3/R)\sin(2\pi 60t)$. The waveforms for the input and output are shown in Figure 3.12.

Note that the sinusoid's phase has indeed shifted; the lowpass filter not only reduced the 60 Hz signal's amplitude, but also shifted its phase by $90\,°$.

3.8 Formal Circuit Methods

In some (complicated) cases, we cannot use the simplification techniques —such as parallel or series combination rules —to solve for a circuit's input-output relation. In previous sections, we wrote *v-i* relations and Kirchoff's laws haphazardly, solving them more on intuition than procedure. We need a formal method that produces a small, easy set of equations that lead directly to the input-output relation we seek. One such technique is the *node method.*

The node method begins by finding all nodes —places where circuit elements attach to each other—in the circuit. One of them we call the *reference node*; the

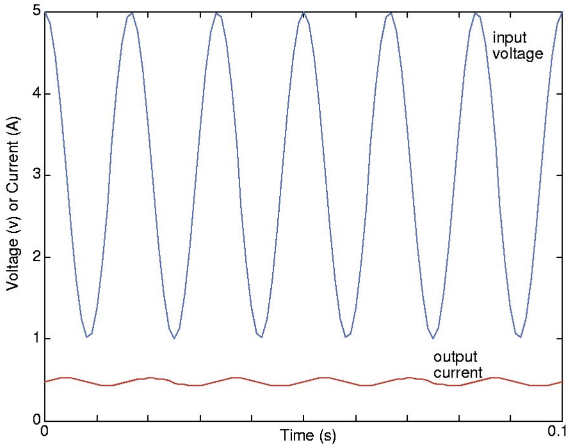
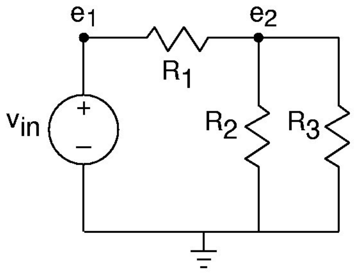


Figure 3.12 Input and output waveforms for the example RL circuit when the element values are $R = 6.28\ \Omega$ and $L = 100$ mH.

choice of reference node is arbitrary, but it is usually chosen to be a point of symmetry or the "bottom" node. For the remaining nodes, define *node voltages* e_n that represent the voltage between the node and the reference. These node voltages constitute the only unknowns; all we need is a sufficient number of equations to solve for them. In our example, we have two node voltages. *The very act of defining node voltages is equivalent to using all the KVL equations at your disposal.* The reason for this simple, but astounding, fact is that a node voltage is uniquely defined regardless of what path you trace between the node and the reference. Because two paths between a node and reference have the same voltage, the sum of voltages around the loop equals zero.



In some cases, a node voltage corresponds exactly to the voltage across a voltage source. In such cases, the node voltage is specified by the source and is *not* an unknown. For example, in our circuit, $e_1 = v_{\text{in}}$; thus, we need only to find one node voltage.

The equations governing the node voltages are obtained by writing KCL equations at each node having an unknown node voltage, using the *v*-*i* relations for each element as you write them. In our example, the only circuit equation is

$$\frac{e_2 - v_{\text{in}}}{R_1} + \frac{e_2}{R_2} + \frac{e_2}{R_3} = 0\ .$$

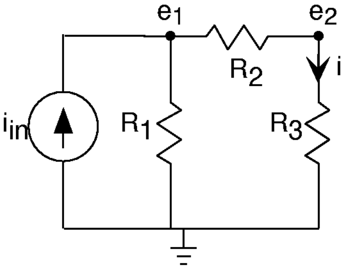
A little reflection reveals that when writing the KCL equations for the sum of currents leaving a node, that node's voltage will *always* appear with a plus sign, and all other node voltages with a minus sign. Systematic application of this procedure makes it easy to write node equations and to check them before solving them. Also remember to check units at this point: Every term should have units of current. In our example, solving for the unknown node voltage is easy:

$$e_2 = \frac{R_2 R_3}{R_1 R_2 + R_2 R_3 + R_1 R_3} v_{\text{in}} \, .$$

Have we really solved the circuit with the node method? Along the way, we have used KVL, KCL, and the *v*-*i* relations. Previously, we indicated that the set of equations resulting from applying these laws is necessary and sufficient. This result guarantees that the node method can be used to "solve" any circuit. One fallout of this result is that we must be able to find any circuit variable given the node voltages and sources. All circuit variables can be found using the *v*-*i* relations and voltage divider. For example, the current through R_3 equals e_2/R_3.

Example

The presence of a current source does not affect the node method greatly; just include it in writing KCL equations as a current *leaving* the node. The circuit has three nodes, requiring us to define two node voltages. The node equations are



$$\text{Node 1: } \frac{e_1}{R_1} - i_{\text{in}} + \frac{e_1 - e_2}{R_2} = 0$$

$$\text{Node 2: } \frac{e_2 - e_1}{R_2} + \frac{e_2}{R_3} = 0$$

Note that the node voltage corresponding to the node we are writing KCL enters with a positive sign, the others with a negative sign, and that the units of each term is amperes. Rewrite these equations in the standard set-of-linear-equations form.

$$e_1 \left(\frac{1}{R_1} + \frac{1}{R_2} \right) - e_2 \frac{1}{R_2} = i_{\text{in}}$$

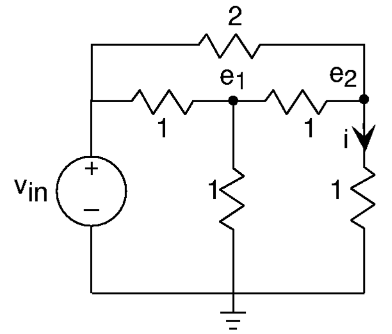
$$-e_1 \frac{1}{R_2} + e_2 \left(\frac{1}{R_2} + \frac{1}{R_3} \right) = 0$$

Solving these equations gives

$$e_1 = \frac{R_2 + R_3}{R_3} e_2$$
$$e_2 = \frac{R_1 R_3}{R_1 + R_2 + R_3} i_{\text{in}}$$

To find the indicated current, we simply use $i = e_2 / R_3$.

Example



In this circuit, we cannot use the series/parallel combination rules: The vertical resistor at node 1 means the two 1 Ω resistors from being in series, and the 2 Ω resistor prevents the two 1 Ω resistors at node 2 from being in series. We really do need the node method to solve this circuit! Despite having six elements, we need only define two node voltages. The node equations are

$$\text{Node 1: } \frac{e_1 - v_{\text{in}}}{1} + \frac{e_1}{1} + \frac{e_1 - e_2}{1} = 0$$
$$\text{Node 2: } \frac{e_2 - v_{\text{in}}}{2} + \frac{e_2}{1} + \frac{e_2 - e_1}{1} = 0$$

Solving these equations yields $e_1 = 2/5\ v_{\text{in}}$ and $e_2 = 5/13\ v_{\text{in}}$. The output current equals $e_2/1 = 5/13\ v_{\text{in}}$. One unfortunate consequence of using the element's numeric values from the outset is that it becomes impossible to check units while setting up and solving equations.

The node method applies without significant modification to RLC circuits if we use complex amplitudes. We rely on the fact that complex amplitudes satisfy KVL, KCL, and impedance-based *v*-*i* relations. In the example circuit, we define complex amplitudes for the input and output variables and for the node voltages. We need only one node voltage here, and its KCL equation is

$$\frac{E - V_{\text{in}}}{R_1} + E \cdot j2\pi f C + \frac{E}{R_2} = 0$$

with the result

$$E = \frac{R_2}{R_1 + R_2 + j2\pi f R_1 R_2 C} V_{\text{in}}$$

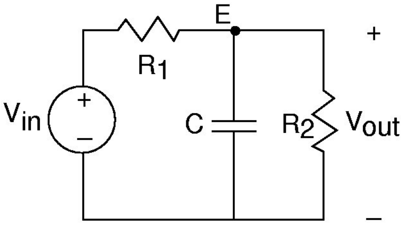


Figure 3.13 Modification of the circuit shown in Figure 3.9 {39} to illustrate the node method and the effect of adding the resistor R_2.

To find the transfer function between input and output voltages, we compute the ratio E/V_{in}. The transfer function's magnitude and angle are

$$|H(f)| = \frac{R_2}{\sqrt{(R_1 + R_2)^2 + (2\pi f R_1 R_2 C)^2}}$$

$$\angle H(f) = -\tan^{-1}\left(\frac{2\pi f R_1 R_2 C}{R_1 + R_2}\right)$$

This circuit differs from the one shown on page {39} in that the resistor R_2 has been added across the input. What effect has it had on the transfer function, which in the original circuit was a lowpass filter having cutoff frequency $f_c = 1/2\pi R_1 C$ (Figure 3.10 {43}). As shown in Figure 3.14, adding the second resistor has two effects: It lowers the gain in the passband (the range of frequencies for which the filter has little effect on the input) and increases the cutoff frequency. When $R_2 = R_1$, as shown on the plot, the passband gain becomes half of the original, and the cutoff frequency increases by the same factor. Thus, adding R_2 provides a "knob" by which we can trade passband gain for cutoff frequency.

Question 3.10 {**249**} *We can change the cutoff frequency* without *affecting passband gain by changing the resistance in the original circuit. Does the addition of the R_2 resistor help in circuit design?*

3.9 Circuit Models for Communication Channels

Maxwell's equations neatly summarize the physics of all electromagnetic phenomena, including circuits, radio, and optic fiber transmission.

$$\boxed{\begin{array}{ll} \nabla \times \vec{E} = -\dfrac{\partial(\mu \vec{H})}{\partial t} & \nabla \cdot (\epsilon \vec{E}) = \rho \\ \nabla \times \vec{H} = \sigma \vec{E} + \dfrac{\partial(\epsilon \vec{E})}{\partial t} & \nabla \cdot \vec{B} = 0 \end{array}}$$

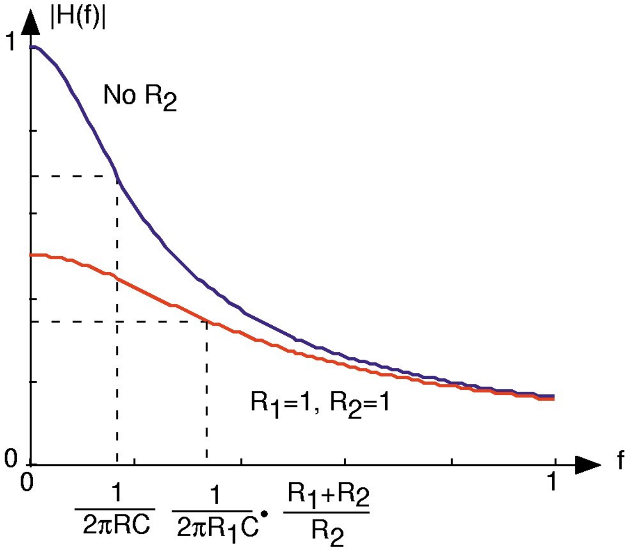


Figure 3.14 Transfer functions of the circuits shown in Figs. 3.9 and 3.13. Here, $R_1 = 1$, $R_2 = 1$, and $C = 1$.

where $\vec{E}$ is the electric field, $\vec{H}$ the magnetic field, ϵ dielectric permittivity, μ magnetic permeability, σ electrical conductivity, ρ charge density, and ∇ represents the gradient operator. Kirchoff's Laws represent special cases of these equations for circuits. We are not going to solve Maxwell's equations here; do bear in mind that a fundamental understanding of communications channels ultimately depends on fluency with Maxwell's equations. Perhaps the most important aspect of them is that they are *linear* with respect to the electrical and magnetic fields. Thus, the fields (and therefore the voltages and currents) resulting from two or more sources will *add.*[*]

The two basic categories for communication channels are *wireline* and *wireless* channels. In wireline channels, transmitter and receiver are connected by a physical conductor. Examples are telephone in the home and cable television. Wireline channels are more intimate than wireless channels: To listen in on a conversation requires that the wire be tapped and the voltage measured. Wireless channels are based on the fact that Maxwell's equations predict that electromagnetic energy will propagate in free space. Applying a voltage to an antenna will cause the signal to radiate in all directions, and a receiving antenna, if located close enough, can monitor the radiation and produce a voltage (ideally) proportional to that transmitted.

---

[*] Nonlinear electromagnetic media do exist. The equations as written here are simpler versions that apply to free-space propagation and conduction in metals. Nonlinear media are becoming increasingly important in optic fiber communications, which are also governed by an elaboration of Maxwell's equations.

3.9.1 Wireline Channels

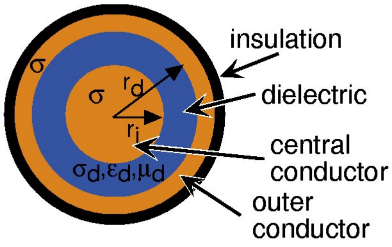


Figure 3.15 Coaxial cable consists of one conductor wrapped around the central conductor. This type of cable supports broader bandwidth signals than twisted pair, and finds use in cable television and Ethernet.

We begin with wireline channels, which were the first used for electrical communications in the mid-nineteenth century for the telegraph. Here, the channel is one of several wires connecting transmitter to receiver. The transmitter simply creates a voltage related to the message signal and applies it to the wire(s). We must have a circuit—a closed path—that supports current flow. In the case of single-wire communications, the earth is used as the current's return path. In fact, the term "ground" for the reference node in circuits originated in single-wire telegraphs. You can imagine that the earth's electrical characteristics are highly variable, and they are. Single-wire metallic channels cannot support high-quality signal transmission having a bandwidth beyond a few hundred Hertz over any appreciable distance.

Consequently, most wireline channels today essentially consist of pairs of conducting wires (Fig. 3.15), and the transmitter applies a message-related voltage across the pair. How these pairs of wires are physically configured greatly affects their transmission characteristics. One example is *twisted pair*, wherein the wires are wrapped about each other. Telephone cables are one example of a twisted pair channel. Another is *coaxial cable*, where a concentric conductor surrounds a central wire with a dielectric material in between. Coaxial cable, fondly called "co-ax" by engineers, is what Ethernet uses as its channel. In either case, wireline channels form a dedicated circuit between transmitter and receiver. As we shall find subsequently, several transmissions can share the circuit by amplitude modulation techniques; commercial cable TV is an example. These information-carrying circuits are designed so that interference from nearby electromagnetic sources is minimized. Thus, by the time signals arrive at the receiver, they are relatively interference- and noise-free.

Both twisted pair and co-ax are examples of *transmission lines*, which all have the circuit model shown in Fig. 3.16 for an infinitesimally small length. This circuit model arises from solving Maxwell's equations for the particular transmission

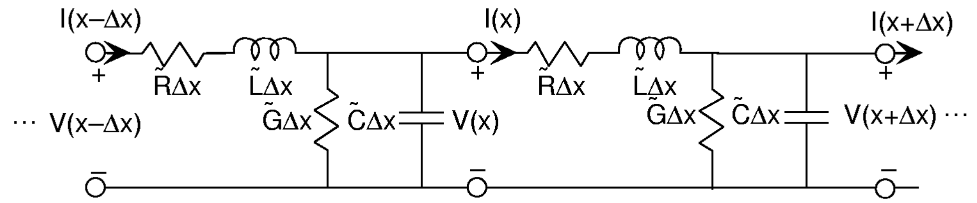


Figure 3.16 The so-called distributed parameter model for two-wire cables has the depicted circuit model structure. Element values depend on geometry and the properties of materials used to construct the transmission line.

line geometry. The series resistance comes from the conductor used in the wires and from the conductor's geometry.* The inductance and the capacitance derive from transmission line geometry, and the parallel conductance from the medium between the wire pair. Note that all the circuit elements have values expressed by the product of a constant times a length; this notation represents that element values here have per-unit-length units. For example, the series resistance $\widetilde{R}$ has units of ohms/meter. For coaxial cable, the element values depend on the inner conductor's radius r_i, the outer radius of the dielectric r_d, the conductivity σ of the conductors, and the conductivity σ_d, dielectric constant ϵ_d, and magnetic permittivity μ_d of the dielectric as

$$\widetilde{R} = \frac{1}{2\pi\delta\sigma}\left(\frac{1}{r_d} + \frac{1}{r_i}\right), \quad \widetilde{C} = \frac{2\pi\epsilon_d}{\ln(r_d/r_i)}, \quad \widetilde{G} = \frac{2\pi\sigma_d}{\ln(r_d/r_i)}, \quad \widetilde{L} = \frac{\mu_d}{2\pi}\ln\frac{r_d}{r_i}$$

For twisted-pair, having a separation d between the conductors that have conductivity σ and common radius r and that are immersed in a medium having dielectric and magnetic properties, the following element values derive.

$$\widetilde{R} = \frac{1}{\pi r\delta\sigma}, \quad \widetilde{C} = \frac{\pi\epsilon}{\cosh^{-1}(d/2r)}, \quad \widetilde{G} = \frac{\pi\sigma}{\cosh^{-1}(d/2r)}, \quad \widetilde{L} = \frac{\mu}{\pi}\left(\frac{\delta}{2r} + \cosh^{-1}\frac{d}{2r}\right)$$

The voltage between the two conductors and the current flowing through them will depend on distance x along the transmission line as well as time. We express this dependence as $v(x,t)$ and $i(x,t)$. When we place a sinusoidal source at one end of the transmission line, these voltages and currents will also be sinusoidal (the transmission line model consists of linear circuit elements). As is customary in

---

[*] Most conductors are cylinders, with current flowing down the cylinder's axis. This static view does not carry over to sinusoidal fields, however. As frequency increases, Maxwell's equations indicate that current flows axially *only* within a shell of depth δ from the conductor's surface. This phenomenon is known as the *skin effect* and δ the skin depth. Quantitatively, $\delta = 1/\sqrt{\pi f\mu\sigma}$.

analyzing linear circuits, we express voltages and currents as the real part of complex exponential signals, and write circuit variables as a complex amplitude —here dependent on distance—times a complex exponential: $v(x,t) = \text{Re}\left[V(x)e^{j2\pi ft}\right]$ and $i(x,t) = \text{Re}\left[I(x)e^{j2\pi ft}\right]$. Using the transmission line circuit model, we find from KCL, KVL, and *v*-*i* relations the equations governing the complex amplitudes.

$$I(x) = I(x-\Delta x) - V(x)(\tilde{G} + j2\pi f\tilde{C})\Delta x \qquad \text{(KCL at center node)}$$
$$V(x) - V(x+\Delta x) = I(x)(\tilde{R} + j2\pi f\tilde{L})\Delta x \qquad (V\text{-}I \text{ relation for } RL \text{ series})$$

Rearranging and taking the limit $\Delta \to 0$ yields the so-called *transmission line equations.*

$$\boxed{\begin{aligned}\frac{dI(x)}{dx} &= -(\tilde{G} + j2\pi f\tilde{C})V(x)\\ \frac{dV(x)}{dx} &= -(\tilde{R} + j2\pi f\tilde{L})I(x)\end{aligned}} \tag{3.3}$$

By combining these equations, we can obtain a single equation that governs how the voltage's or the current's complex amplitude changes with position along the transmission line. Taking the derivative of the second equation and plugging the first equation into the result yields the equation governing the voltage.

$$\frac{d^2V(x)}{dx^2} = (\tilde{G} + j2\pi f\tilde{C})(\tilde{R} + j2\pi f\tilde{L})V(x)$$

This equation's solution is

$$V(x) = V_+e^{-\gamma x} + V_-e^{+\gamma x}\ .$$

Calculating its second derivative and comparing the result with our equation for the voltage can check this solution.

$$\frac{d^2V(x)}{dx^2} = \gamma^2 \underbrace{\left[V_+e^{-\gamma x} + V_-e^{+\gamma x}\right]}_{V(x)}$$

Our solution works so long as the quantity γ satisfies

$$\boxed{\gamma = \pm\sqrt{(\tilde{G} + j2\pi f\tilde{C})(\tilde{R} + j2\pi f\tilde{L})} = \pm[a(f) + jb(f)]}\ . \tag{3.4}$$

Thus, γ depends on frequency, and we express it in terms of real and imaginary parts as indicated. The quantities V_+ and V_- are constants determined by the source and physical considerations. For example, let the spatial origin be the middle of the transmission line model Figure 3.16 $\{56\}$. Because the circuit model contains simple circuit elements, physically possible solutions for voltage amplitude cannot increase with distance along the transmission line. Expressing γ in terms of its real and imaginary parts in our solution shows that such increases are a (mathematical) possibility.

$$V(x) = V_{+}e^{-(a+jb)x} + V_{-}e^{+(a+jb)x}$$

The voltage cannot increase without limit; because $a(f)$ is always positive, we must segregate the solution for negative and positive x. The first term will increase exponentially for $x < 0$ unless $V_{+} = 0$ in this region; a similar result applies to V_{-} for $x > 0$. These physical constraints give us a cleaner solution.

$$V(x) = \begin{cases} V_{+}e^{-(a+jb)x}, & x > 0 \\ V_{-}e^{+(a+jb)x}, & x < 0 \end{cases}$$

This solution suggests that voltages (and currents too) will decrease *exponentially* along a transmission line. The *space constant*, also known as the *attenuation constant*, is the distance over which the voltage decreases by a factor of $1/e$. It equals $a(f)$, depends on frequency, and is expressed by manufacturers in units of dB/m.

The presence of the imaginary part of γ, $b(f)$, also provides insight into how transmission lines work. Because the solution for $x > 0$ is proportional to e^{-jbx}, we know that the voltage's complex amplitude will *vary sinusoidally in space*. The complete solution for the voltage has the form $v(x,t) = \text{Re}\left[V_{+}e^{-ax}e^{j(2\pi ft-bx)}\right]$. The complex exponential portion has the form of a *propagating wave*. If we could take a snapshot of the voltage (take its picture at $t = t_1$), we would see a sinusoidally varying waveform along the transmission line. One period of this variation, known as the *wavelength*, equals

$$\lambda = \frac{2\pi}{b} \, .$$

If we were to take a second picture at some later time $t = t_2$, we would also see a sinusoidal voltage. Because

$$2\pi ft_2 - bx = 2\pi f\big(t_1 + (t_2 - t_1)\big) - bx = 2\pi ft_1 - b\left(x - \frac{2\pi f}{b}(t_2 - t_1)\right) \, ,$$

the second waveform appears to be the first one, but delayed—shifted to the right—in space. Thus, the voltage appeared to move to the right with a speed equal to $2\pi f/b$ (assuming $b > 0$). We denote this *propagation speed* by c, and it equals

$$c = \left| \frac{2\pi f}{\text{Im}\left[\sqrt{(\widetilde{G} + j2\pi f\widetilde{C})(\widetilde{R} + j2\pi f\widetilde{L})}\right]} \right| .$$

In the high-frequency region where $2\pi f\widetilde{L} \gg \widetilde{R}$ and $2\pi f\widetilde{C} \gg \widetilde{G}$, the quantity under the radical simplifies to $-4\pi^2 f^2 \widetilde{L}\widetilde{C}$, and we find the propagation speed to be

$$\lim_{f\to\infty} c = \frac{1}{\sqrt{\widetilde{L}\widetilde{C}}} .$$

For typical coaxial cable, this propagation speed is a fraction $(1/3 - 2/3)$ of the speed of light.

Question 3.11 $\{$**249**$\}$ *Find the propagation speed in terms of physical parameters for both the coaxial cable and twisted pair examples.*

By using the second of the transmission line equations (3.3) $\{57\}$, we can solve for the current's complex amplitude. Considering the spatial region $x > 0$, for example, we find that

$$\frac{dV(x)}{dx} = -\gamma \underbrace{V_+ e^{-\gamma x}}_{V(x)} = -(\widetilde{R} + j2\pi f\widetilde{L})I(x) ,$$

which means that the ratio of voltage and current complex amplitudes does not depend on distance.

$$\frac{V(x)}{I(x)} = \sqrt{\frac{\widetilde{R} + j2\pi f\widetilde{L}}{\widetilde{G} + j2\pi f\widetilde{C}}} = Z_0$$

The quantity Z_0 is known as the transmission line's *characteristic impedance*. Note that when the signal frequency is sufficiently high, the characteristic impedance is real, which means the transmission line appears resistive in this high-frequency regime.

$$\lim_{f\to\infty} Z_0 = \sqrt{\frac{\widetilde{L}}{\widetilde{C}}}$$

Typical values for characteristic impedance are 50 and 75 Ω.

A related transmission line is the optic fiber. Here, the electromagnetic field is light, and it propagates down a cylinder of glass. In this situation, we don't have two conductors —in fact we have none —and the energy is propagating in what corresponds to the dielectric material of the coaxial cable. Optic fiber communication has exactly the same properties as other transmission lines: Signal strength decays exponentially according to the fiber's space constant and propagates at some speed less than light would in free space. From the encompassing view of Maxwell's equations, the only difference is the electromagnetic signal's frequency. Because no electric conductors are present and the fiber is protected by an opaque "insulator," optic fiber transmission is interference-free.

Question 3.12 {**249**} *From tables of physical constants, find the frequency of a sinusoid in the middle of the visible light range. Compare this frequency with that of a mid-frequency cable television signal.*

To summarize, we use transmission lines for high-frequency wireline signal communication. In wireline communication, we have a direct, physical connection—a circuit—between transmitter and receiver. When we select the transmission line characteristics and the transmission frequency so that we operate in the high-frequency regime, signals are not filtered as they propagate along the transmission line: The characteristic impedance is real and all the signal's components at various frequencies propagate at the same speed. Transmitted signal amplitude does decay exponentially along the transmission line. Note that in the high-frequency regime that the space constant is approximately zero, which means the attenuation is quite small.

Question 3.13 {**249**} *What is the limiting value of the space constant in the high frequency regime?*

3.9.2 Wireless Channels

Wireless channels exploit the prediction made by Maxwell's equation that electromagnetic fields propagate in free space like light. When a voltage is applied to an antenna, it creates an electromagnetic field that propagates in all directions (although antenna geometry affects how much power flows in any given direction) that induces electric currents in the receiver's antenna. Antenna geometry determines how energetic a field a voltage of a given frequency creates. In general terms, dominant factor is the relation of the antenna's size to the field's wavelength. The fundamental equation relating frequency and wavelength for a propagating wave is

$$\boxed{\lambda f = c}\ .$$

Thus, wavelength and frequency are inversely related: High frequency corresponds to small wavelengths. For example, a 1 MHz electromagnetic field has a wavelength of 300 m. Antennas having a size or distance from the ground comparable to the wavelength radiate fields most efficiently. Consequently, the lower the frequency the bigger the antenna must be. Because most information signals are baseband signals, having spectral energy at low frequencies, they must be modulated to higher frequencies to be transmitted over wireless channels.

For most antenna-based wireless systems, how the signal diminishes as the receiver mves further from the transmitter derives by considering how radiated power changes with distance from the transmitting antenna. An antenna radiates a given amount of power into free space, and ideally this power propagates without loss in all directions. Considering a sphere centered at the transmitter, the total power, which is found by integrating the radiated power over the surface of the sphere, must be constant regardless of the sphere's radius. This requirement results from the conservation of energy. Thus, if $P(d)$ represents the power integrated with respect to direction at a distance d from the antenna, the total power will be $P(d) \cdot 4\pi d^2$. For this quantity to be a constant, we must have

$$\boxed{P(d) \propto \frac{1}{d^2}} \tag{3.5a}$$

which means that the received signal amplitude A_R must be proportional to the transmitter's amplitude A_T and inversely related to distance from the transmitter.

$$\boxed{A_R = \frac{kA_T}{d}} \tag{3.5b}$$

for some value of the constant k. Thus, the further from the transmitter the receiver is located, the weaker the received signal. Whereas the attenuation found in wireline channels can be controlled by physical parameters and choice of transmission frequency, the inverse-distance attenuation found in wireless channels persists across all frequencies.

The speed of propagation is governed by the dielectric constant ϵ_0 and magnetic permeability μ_0 of free space.

$$c = \frac{1}{\sqrt{\mu_0 \epsilon_0}} = 3 \times 10^8 \text{m/s}$$

Known familiarly as the speed of light, it sets an upper limit on how fast signals can propagate from one place to another. Because signals travel at a finite speed, a receiver senses a transmitted signal only after a time delay directly related to the propagation speed: $\Delta t = d/c$. At the speed of light, a signal travels across the United States in 16 ms, a reasonably small time delay. If a lossless (zero space constant) coaxial cable connected the East and West coasts, this delay would be two to three times longer because of the slower propagation speed.

3.10 Electronics

So far in this chapter, we have analyzed *electrical* circuits: The source signal has more power than the output variable, be it a voltage or a current. Power has not been explicitly defined, but no matter. Resistors, inductors, and capacitors as individual elements certainly provide no power gain, and circuits of them will not magically do so either. Such circuits are termed electrical in distinction to those that do: *electronic* circuits. Providing power gain, such as your stereo reading a CD and producing sound, is accomplished by semiconductor circuits that contain transistors. The basic idea of the transistor is to let the weak input signal modulate a strong current provided by a source of electrical power —the power supply —to produce a more powerful signal. A physical analogy is a water faucet: By turning the faucet back and forth, the water flow varies accordingly, and has much more power than expended in turning the handle. The waterpower results from the static pressure of the water in your plumbing created by the water utility pumping the water up to your local water tower. The power supply is like the water tower, and the faucet is the transistor, with the turning achieved by the input signal. Just as in this analogy, a power supply is a source of constant voltage as the water tower is supposed to provide a constant water pressure.

We will not discuss the transistor here. A much more convenient device for providing gain (and other useful features as well) is the *operational amplifier*, also known as the *op-amp*. An op-amp is an integrated circuit (a complicated circuit involving several transistors constructed on a chip) that provides a large voltage gain *if* you attach the power supply. We can model the op-amp with a new circuit element: the dependent source.

3.10.1 Dependent Sources

A *dependent source* is either a voltage or current source whose value is proportional to some other voltage or current in the circuit. Thus, there are four different kinds of dependent sources; to describe an op-amp, we need a voltage-dependent voltage source. However, the standard circuit-theoretic model for a transistor contains a current-dependent current source. Dependent sources do not serve as inputs to a circuit like independent sources. They are used to model *active* circuits: those containing electronic elements. The RLC circuits we have been considering so far are known as passive circuits.

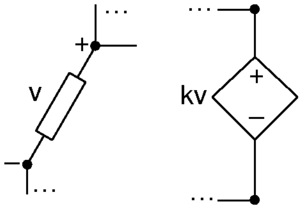


Figure 3.17 Of the four possible dependent sources, depicted is a voltage-dependent voltage source in the context of a generic circuit.

Fig. 3.18 shows the circuit symbol for the op-amp and its equivalent circuit in terms of a voltage-dependent voltage source. Here, the output voltage equals an amplified version of the difference of node voltages appearing across its inputs. The dependent source model portrays how the op-amp works quite well. As in most active circuit schematics, the power supply is not shown, but must be present or the circuit model to be accurate. Most operational amplifiers require both positive and negative supply voltages for proper operation.

Because dependent sources cannot be described as impedances, and because the dependent variable cannot "disappear" when you apply parallel/series combining rules, circuit simplifications such as current and voltage divider should not be applied in most cases. Analysis of circuits containing dependent sources essentially requires use of formal methods, like the node method. Using the node method for such circuits is not difficult, with node voltages defined across the source treated as if they were known (as with independent sources). Consider the circuit shown on the left in Figure 3.19. Note that the op-amp is placed in the circuit "upside-down," with its inverting input at the top and serving as the only input. As we explore op-amps in more detail in the next section, this configuration will appear again and again and is shown to be quite useful. To determine how the output voltage is re-

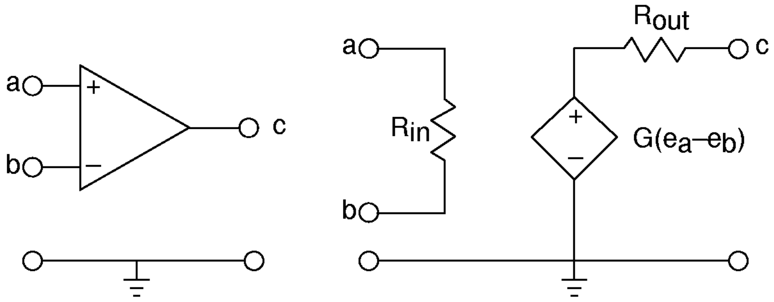


Figure 3.18 The op-amp has four terminals to which connections can be made. Inputs attach to nodes a and b, and the output is node c. As the circuit model on the right shows, the op-amp serves as an amplifier for the difference of the input node voltages.

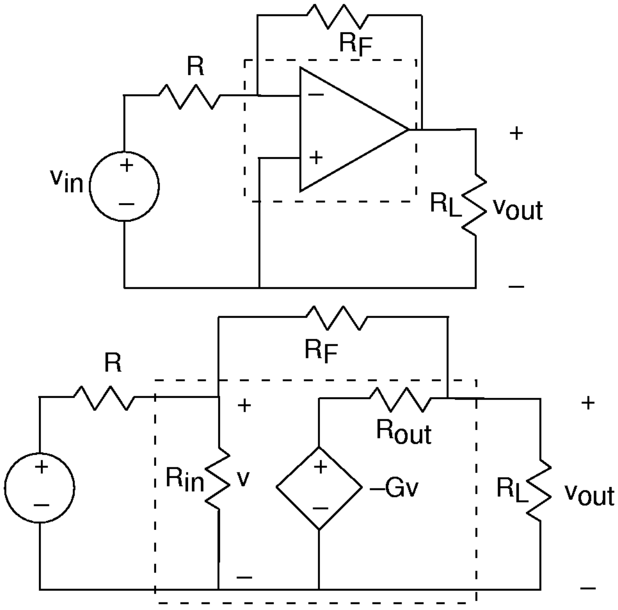


Figure 3.19 The top circuit depicts an op-amp in a feedback amplifier configuration. On the bottom is the equivalent circuit, and integrates the op-amp circuit model into the circuit.

lated to the input voltage, we apply the node method. Only two node voltages —v and v_{out} —need be defined; the remaining nodes are across sources or serve as the reference. The node equations are

$$\frac{v - v_{\text{in}}}{R} + \frac{v}{R_{\text{in}}} + \frac{v - v_{\text{out}}}{R_{\text{F}}} = 0$$

$$\frac{v_{\text{out}} - (-Gv)}{R_{\text{out}}} + \frac{v_{\text{out}} - v}{R_{\text{F}}} + \frac{v_{\text{out}}}{R_{\text{L}}} = 0$$

Note that no special considerations were used in applying the node method to this dependent-source circuit. Solving these to learn how v_{out} relates to v_{in} yields

$$\left[\left(\frac{R_{\text{F}}R_{\text{out}}}{R_{\text{out}} - GR_{\text{F}}}\right)\left(\frac{1}{R_{\text{out}}} + \frac{1}{R_{\text{in}}} + \frac{1}{R_{\text{L}}}\right)\left(\frac{1}{R} + \frac{1}{R_{\text{in}}} + \frac{1}{R_{\text{F}}}\right) - \frac{1}{R_{\text{F}}}\right] v_{\text{out}} = \frac{1}{R} v_{\text{in}} . \tag{3.6}$$

This expression represents the general input-output relation for this circuit, known as the *standard feedback configuration*. Once we learn more about op-amps, in particular what its typical element values are, the expression will simplify greatly. Do note that the units check, and that the parameter G of the dependent source is a dimensionless gain.

3.10.2 Operational Amplifiers

Op-amps not only have the circuit model shown in Figure 3.18, their element values are very special.

- The **input resistance** R_{in} is typically **large**, on the order of $1\ \text{M}\Omega$.
- The **output resistance** R_{out} is **small**, usually less than $100\ \Omega$.
- The **voltage gain** G is **large**, exceeding 10^5.

The large gain catches the eye; it suggests that an op-amp could turn a 1 mv input signal into a 100 v one. If you were to build such a circuit —attaching a voltage source to node a, attaching node b to the reference, and looking at the output —you would be disappointed. In dealing with electronic components, you cannot forget the unrepresented but needed power supply. *It is impossible for electronic components to yield voltages that exceed those provided by the power supply or for them to yield currents that exceed the power supply's rating.* Typical power supply voltages required for op-amp circuits is ± 15 v. Attaching the 1 mv signal not only would fail to produce a 100 v signal, the resulting waveform would be severely distorted. While a desirable outcome if you are a rock&roll aficionado, high-quality stereos should not distort signals. Another consideration is designing circuits with op-amps is that these element values are typical: Careful control of the gain can only be obtained by choosing a circuit so that its element values dictate the resulting gain, which must be smaller than that provided by the op-amp.

The feedback configuration shown in Figure 3.19 is the most common op-amp circuit for obtaining what is known as an *inverting amplifier.* Eq. 3.6 $\{64\}$ provides the exact input-output relationship. In choosing element values with respect to op-amp characteristics, we can simplify the expression dramatically.

- Make the load resistance R_{L} much larger than R_{out}. This situation drops the term $1/R_{\text{L}}$ from the second factor of Eq. (3.6).
- Make the resistor R smaller than R_{in}, which means that the $1/R_{\text{in}}$ term in the third factor is negligible.

With these two design criteria, the expression (3.6) becomes

$$\left[\left(\frac{R_{\text{F}}+R_{\text{out}}}{R_{\text{out}}-GR_{\text{F}}}\right)\left(\frac{1}{R}+\frac{1}{R_{\text{F}}}\right)-\frac{1}{R_{\text{F}}}\right]v_{\text{out}}=\frac{1}{R}v_{\text{in}}\ .$$

Because the gain is large and the resistance R_{out} is small, the first term becomes $-1/G$, leaving us with

$$\left[-\frac{1}{G}\left(\frac{1}{R}+\frac{1}{R_{\text{F}}}\right)-\frac{1}{R_{\text{F}}}\right]v_{\text{out}}=\frac{1}{R}v_{\text{in}}\ .$$

- If we select the values of R_{F} and R so that $R_{\text{F}} \ll GR$, this factor will no longer depend on the op-amp's inherent gain, and it will equal $-\frac{1}{R_{\text{F}}}$.

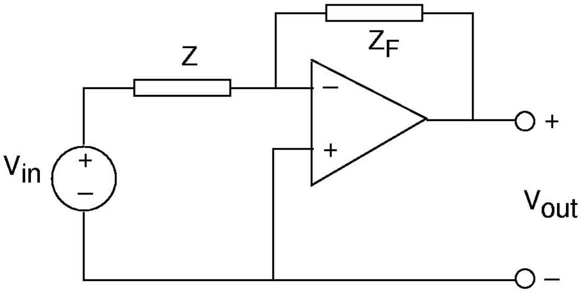
Under these conditions, we obtain the classic input-output relationship for the op-amp-based inverting amplifier.

$$\boxed{v_{\text{out}} = -\frac{R_{\text{F}}}{R} v_{\text{in}}}$$

Consequently, the gain provided by our circuit is entirely determined by our choice of the feedback resistor R_{F} and the input resistor R. It can even be less than one, but cannot exceed the op-amp's inherent gain and should not produce such large outputs that distortion results (remember the power supply!). Interestingly, note that this relationship does not depend on the load resistance. This effect occurs because we use load resistances large compared to the op-amp's output resistance. Thus observation means that, if careful, we can place op-amp circuits in cascade, *without* incurring the effect of succeeding circuits changing the behavior (transfer function) of previous ones; see Problem 3.19 $\{80\}$.

As long as design requirements are met, the input-output relation for the inverting amplifier also applies when the feedback and input circuit elements are impedances (resistors, capacitors, and inductors).

$$\frac{V_{\text{out}}}{V_{\text{in}}} = -\frac{Z_{\text{F}}}{Z}$$



Example

Let's design an op-amp circuit that functions as a lowpass filter. We want the transfer function between the output and input voltage to be

$$H(f) = \frac{K}{1 + j2\pi f/f_c},$$

where K equals the passband gain and f_c is the cutoff frequency. Let's assume that the inversion (negative gain) does not matter. With the transfer function of the above op-amp circuit in mind, let's consider some choices.

- $Z_{\text{F}} = K$, $Z = 1 + j2\pi f/f_c$.
  This choice means the feedback impedance is a resistor and that the input

impedance is a series combination of an inductor and a resistor. In circuit design, we try to avoid inductors because they are physically bulkier than capacitors.

- $Z_\mathrm{F} = 1/(1 + j2\pi f/f_c)$, $Z = 1/K$.
  Consider the reciprocal of the feedback impedance (its admittance): $Z_\mathrm{F}^{-1} = 1 + j2\pi f/f_c$. Thus, the admittance is a sum of admittances, suggesting the parallel combination of a resistor (value $= 1\ \Omega$) and a capacitor (value $= 1/f_c$ F). We have the right idea, but the values (like $1\ \Omega$) aren't right. Consider the general RC parallel combination; its admittance is $\frac{1}{R_\mathrm{F}} + j2\pi fC$. Letting the input resistance equal R, the transfer function of the op-amp inverting amplifier now is

$$H(f) = -\frac{R_\mathrm{F}/R}{1 + j2\pi f R_\mathrm{F} C}\ .$$

Thus, we have the gain equal to R_F/R and the cutoff frequency $1/R_\mathrm{F}C$.

Thus, creating a specific transfer function with op-amps does not have a unique answer. As opposed to design with passive circuits, electronics is more flexible (a cascade of circuits can be built so that each has little effect on the others) and gain (increase in power and amplitude) can result. To complete our example, let's assume we want a lowpass filter that emulates what the telephone companies do. Signal transmitted over the telephone have an upper frequency limit of about 3 kHz. For the second design choice, we require $R_\mathrm{F}C = 3.3 \times 10^{-4}$. Thus, many choices for resistance and capacitance values are possible. A 1 μF capacitor and a 330 Ω resistor, 10 nF and 33 kΩ, and 10 pF and 33 MΩ would all theoretically work. Let's also desire a voltage gain of ten: $R_\mathrm{F}/R = 10$, which means $R = R_\mathrm{F}/10$. Recall that we must have $R \ll R_\mathrm{in}$. As the op-amp's input impedance is about 1 MΩ, we don't want R too large, and this requirement means that the last choice for resistor/capacitor values won't work. We also need to ask for less gain than the op-amp can provide itself. Because the feedback "element" is an impedance (a parallel resistor capacitor combination), we need to examine the gain requirement more carefully. We must have $|Z_\mathrm{F}|/R \ll 10^5$ for all frequencies of interest. Thus, $(R_\mathrm{F}/|1 + j2\pi f R_\mathrm{F} C|)/R \ll 10^5$. As this impedance decreases with frequency, the design specification of $R_\mathrm{F}/R = 10$ means that this criterion is easily met. Thus, the first two choices for the resistor and capacitor values (as well as many others in this range) will work well. Additional considerations like parts cost might enter into the picture. Unless you have a high-power application (this isn't one) or ask for high-precision components, costs don't depend heavily on component values

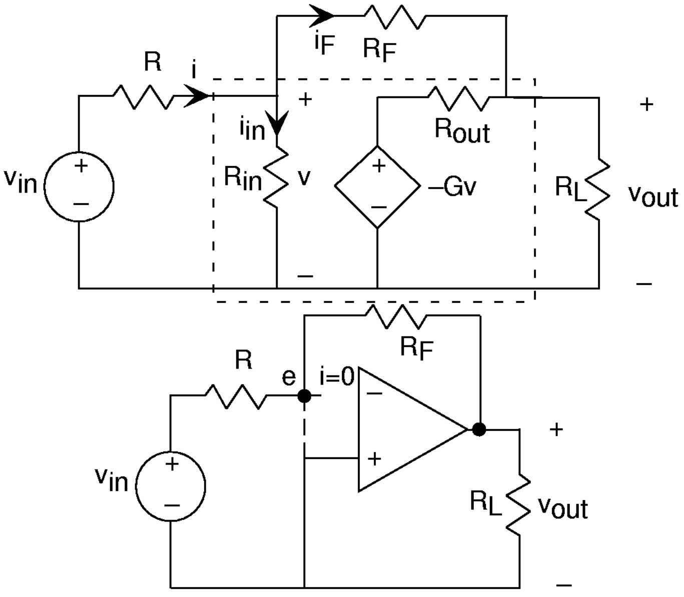


Figure 3.20

as long as you stay close to standard values. For resistors, having values $r \times 10^{d}$, easily obtained values of r are 1, 1.4, 3.3, 4.7, and 6.8, and the decades span $0 - 8$.

Question 3.14 {**250**} *What is special about the resistor values; why these rather odd-appearing values for r?*

When we meet op-amp design specifications, we can simplify our circuit calculations greatly, so much so that we don't need the op-amp's circuit model to determine the transfer function. Let's return to our inverting amplifier. When we take advantage of the op-amp's characteristics —large input impedance, large gain, and small output impedance* —we note the two following important facts.

- The current i_{in} must be very small.
  The voltage produced by the dependent source is 10^{5} times the voltage v. Thus, the voltage v must be small, which means that $i_{\text{in}} = v/R_{\text{in}}$ must be tiny. For example, if the output is about 1 v, the voltage $v = 10^{-5}$ v making the current $i_{\text{in}} = 10^{-11}$ a. Consequently, we can ignore i_{in} in our calculations and assume it to be zero.

---

*Although the op-amp circuit model contains only resistors, we still use the word impedance to describe their values. A resistor is a special case of an impedance!

- Because of this assumption—essentially no current flow through R_{in}—the voltage v must also be essentially zero. This means that in op-amp circuits, the voltage across the op-amp's input is basically zero.

Armed with these approximations, let's return to our original circuit as shown in Figure 3.20 {68}. The node voltage e is essentially zero, meaning that it is essentially tied to the reference node. Thus, the current through the resistor R equals v_{in}/R. Furthermore, the feedback resistor appears in parallel with the load resistor. The voltage across the output resistor thus equals the negative of that across the feedback resistor. Because the current going into the op-amp is zero, all of the current flowing through R flows through the feedback resistor ($i_F = i$)! Consequently, the voltage across it is $v_{\text{in}}R_F/R$ and we have v_{out}. Using this approach makes analyzing new op-amp circuits much easier. Do check to make sure the results you obtain are consistent with the assumptions.

Example

Let's try this analysis technique on a simple extension of the inverting amplifier

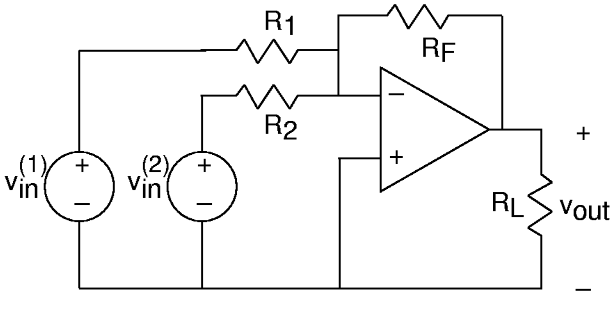


Figure 3.21 Two-source, single-output op-amp circuit example.

configuration shown in Figure 3.21 {69}. If either of the source-resistor combinations were not present, the inverting amplifier remains, and we know that transfer function. By superposition, we know that the input-output relation is

$$v_{\text{out}} = -\frac{R_F}{R_1}v_{\text{in}}^{(1)} - \frac{R_F}{R_2}v_{\text{in}}^{(2)} .$$

When we start from scratch, the node joining the three resistors is at the same potential as the reference and the sum of currents flowing into that node are zero. Thus, the current flowing to the right in the resistor R_F is $v_{\text{in}}^{(1)}/R_1 + v_{\text{in}}^{(2)}/R_2$. Because the feedback resistor is essentially in parallel with the load resistor, the voltages across them are equal. In this way, we obtain the input-output relation given above.

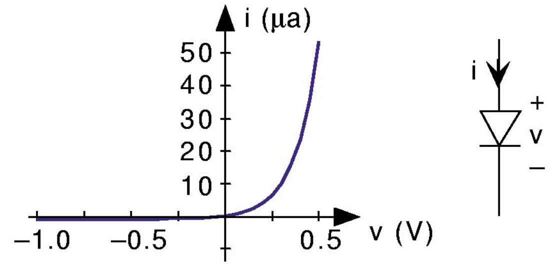


Figure 3.22 *v-i* relation and schematic symbol for the diode. Here, the diode parameters were room temperature and $I_0 = 1\ \mu$a.

What utility would this circuit have? Can the basic notion of the circuit be extended without bound?

3.11 Nonlinear Circuit Elements

The resistor, capacitor, and inductor are linear circuit elements in that their *v-i* relations are linear in the mathematical sense. Voltage and current sources are (technically) nonlinear devices: stated simply, doubling the current through a voltage source does not double the voltage. A more blatant, and very useful, nonlinear circuit element is the diode. Its input-output relation has an exponential form.

$$i(t) = I_0 \left(e^{\frac{q}{kT}v(t)} - 1\right)$$

Here, the quantity q represents the charge of a single electron in coulombs, k is Boltzmann's constant, and T is the diode's temperature in $°K$. At room temperature, the ratio $kT/q = 25$ mv. The constant I_0 is the leakage current, and is usually very small. Viewing what this *v-i* relation in Figure 3.22, the nonlinearity becomes obvious. When the voltage is positive, current flows easily through the diode. This situation is known as *forward biasing*. When we apply a negative voltage, the current is quite small, and equals I_0, known as the *leakage* or *reverse-bias* current. A less detailed model for the diode has any positive current flowing through diode when it is forward biased, and no current when negative biased. Note that the diode's schematic symbol looks like an arrowhead; the direction of current flow corresponds to the direction the arrowhead points.

Because of the diode's nonlinear nature, we *cannot* use impedances nor series/parallel combination rules to analyze circuits containing them. The reliable node method can always be used; it only relies on KVL for its application, and KVL is a statement about voltage drops around a closed path regardless of the path's elements. Thus, for this simple circuit we have

$$\frac{v_{\text{out}}}{R} = I_0\left(e^{\frac{q}{kT}(v_{\text{in}}-v_{\text{out}})}-1\right) . \tag{3.7}$$

This equation *cannot* be solved in closed form. We must understand what is going on from basic principles, using computational and graphical aids. As an approximation, when v_{in} is positive, current flows through the diode so long as the voltage v_{out} is smaller than v_{in} (so the diode is forward biased). If the source is negative or v_{out} "tries" to be bigger than v_{in}, the diode is reverse-biased, and the reverse-bias current flows through the diode. Thus, at this level of analysis, positive input voltages result in positive output voltages with negative ones resulting in $v_{\text{out}} = -RI_0$.

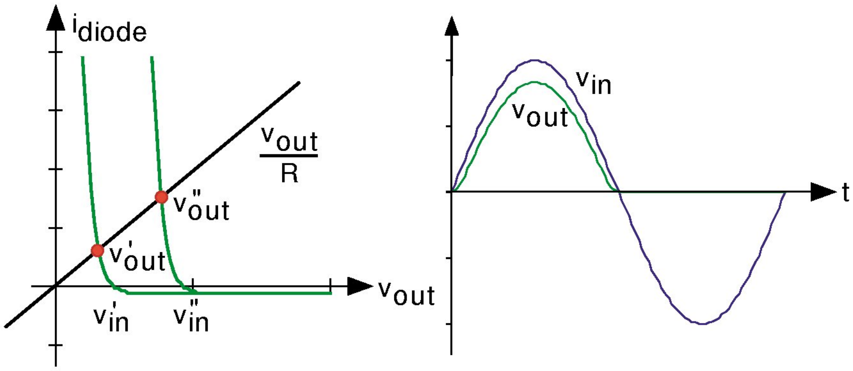
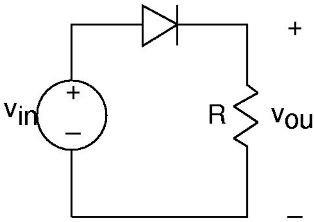
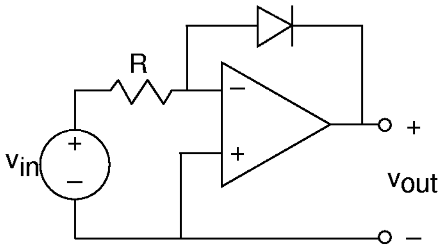


Figure 3.23



In more detail, we need to detail the exponential nonlinearity to determine how the circuit distorts the input voltage waveform. We can of course numerically solve (3.7) to determine the output voltage when the input is a sinusoid. To learn more, let's express this equation graphically. We plot each term as a function of v_{out} for various values of the input voltage v_{in}; where they intersect gives us the output voltage. The left side, the current through the output resistor, does not depend vary itself with v_{in}, and thus we have a fixed straight line. As for the right side, which expresses the diode's *v*-*i* relation, the point at which the curve crosses the v_{out} axis gives us the value of v_{in}. Clearly, the two curves will always intersect just once for any value of v_{in}, and for positive $v_{\text{in}}$the intersection occurs at a value for v_{out}*smaller* than v_{in}. This reduction is smaller if the straight line has a shallower slope, which corresponds to using a bigger output resistor. For negative v_{in}, the diode is reverse-biased and the output voltage equals $-RI_0$.

What utility might this simple circuit have? The diode's nonlinearity cannot be escaped here, and the clearly evident distortion must have some practical application if the circuit were to be useful. This circuit, known as a *half-wave rectifier*, is present in virtually every AM radio twice and each serves very different functions! Why we will have to learn about later.



Here is another circuit involving a diode that is actually simpler to analyze than the previous one. We know that the current through the resistor must equal that through the diode. Thus, the diode's current is proportional to the input voltage. As the voltage across the diode is related to the logarithm of its current, we see that the input-output relation is

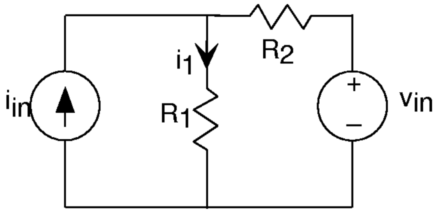
$$v_{\text{out}} = -\frac{kT}{q}\ln\left(\frac{v_{\text{in}}}{RI_0}+1\right) .$$

Clearly, the name *logarithmic amplifier* is justified for this circuit.

Problems

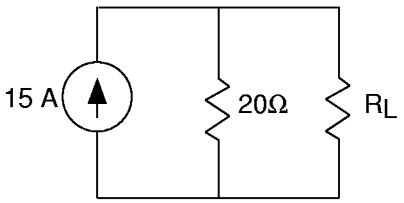
3.1 Solving Simple Circuits

(a) Write the set of equations that govern this circuits behavior.



(b) Solve these equations for i_1: In other words, express this current in terms of element and source values by eliminating non-source voltages and currents.

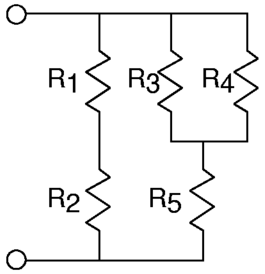
(c) For the following circuit, find the value for R_L that results in a current of 5 A passing through it.



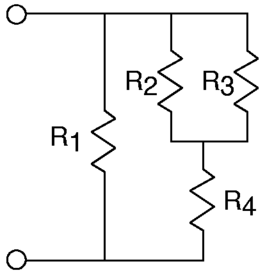
3.2 Equivalent Resistance

For each of the following circuits, find the equivalent resistance using series and parallel combination rules.

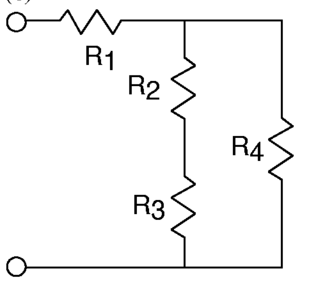
(a)



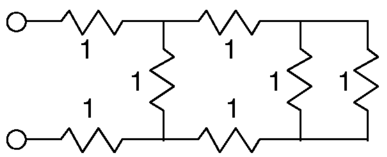
(b)



(c)



(d)

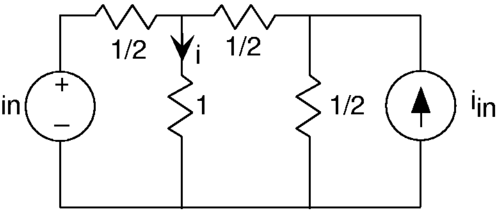


(e) Calculate the conductance seen at the terminals for circuit (c) in terms of each element's conductance. Compare this equivalent conductance formula with the equivalent resistance formula you found for circuit (b). How is the circuit in (c) derived from the one given in (b)?

3.3 Superposition Principle

One of the most important consequences of circuit laws is the Superposition Principle: The current or voltage defined for any element equals the sum of the currents or voltages produced in the element by the independent sources. This Principle has important consequences in simplifying the calculation of circuit variables in multiple source circuits.

(a) For the depicted circuit, find the indicated current using any technique you like. (You should choose the simplest.)

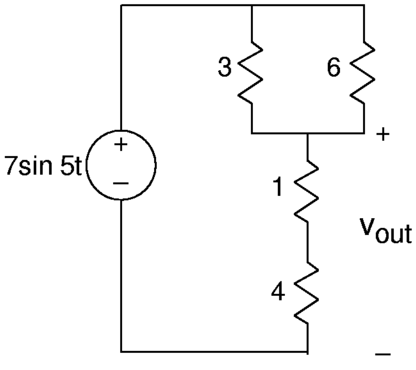


(b) You should have found that the current i is a linear combination of the two source values: $i = C_1 v_{\text{in}} + C_2 i_{\text{in}}$. This result means that we can think of the current as a superposition of two components, each of which is due to a source. We can find each component by setting the other sources to zero. Thus, to find the voltage source component, you can set the current source to zero (an open circuit) and use the usual tricks. To find the current source component, you would set the voltage source to zero (a short circuit) and find the resulting current. Calculate the total current i using the Superposition Principle. Is applying the Superposition Principle easier than the technique you used in part (a)?

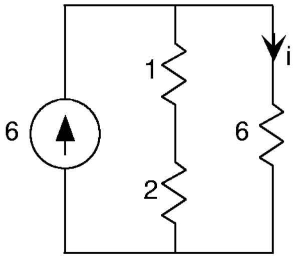
3.4 Current and Voltage Divider

Use current or voltage divider rules to calculate the indicated circuit variable.

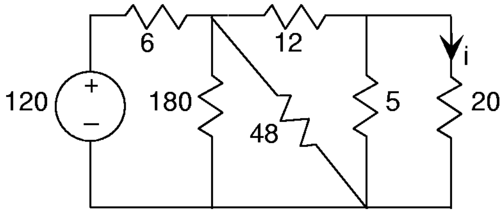
(a)



(b)



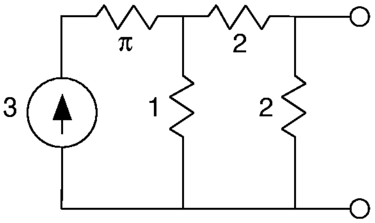
(c)



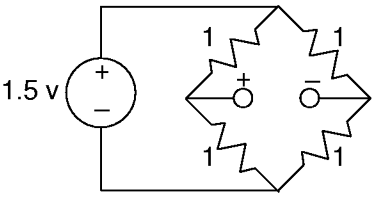
3.5 Thévenin and Norton Equivalents

Find the Thévenin and Norton equivalent circuits for the following circuits.

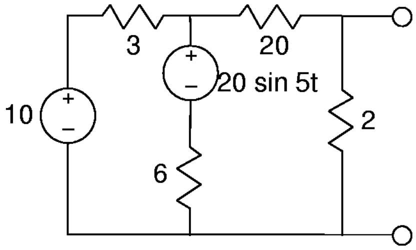
(a)



(b)



(c)

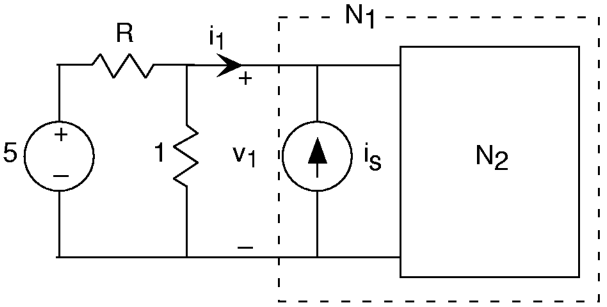

3
20
+
−
20 sin 5t
10
+
−
2
6

3.6 Detective Work

In the depicted circuit, the circuit N_1 has the *v*-*i* relation $v_1 = 3i_1 + 7$ when $i_s = 2$.

(a) Find the Thévenin equivalent circuit for circuit N_2.

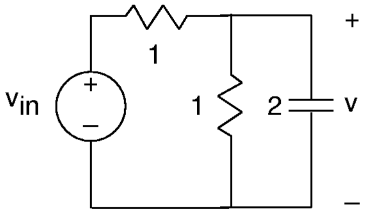
(b) With $i_s = 2$, determine R such that $i_1 = -1$


N1
R
i1
+
5
+
−
1
v1
−
is
N2

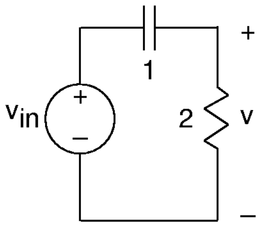
3.7 Transfer Functions

Find the transfer function relating the complex amplitudes of the indicated variable and the source. Plot the magnitude and phase of the transfer function.

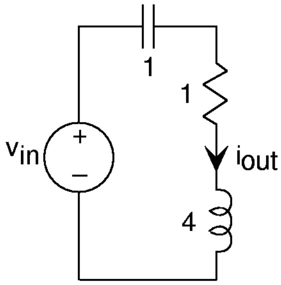
(a)



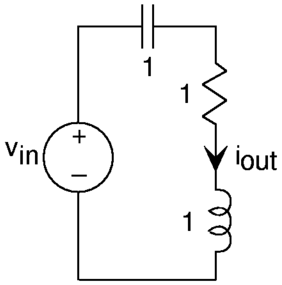
(b)



(c)


1
1
v_{in}
+
−
i_{out}
4

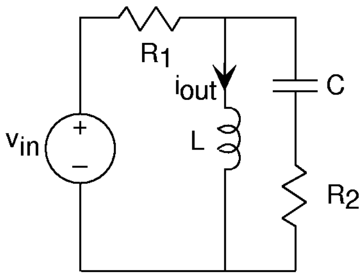
(d)


1
1
v_{in}
+
−
i_{out}
1

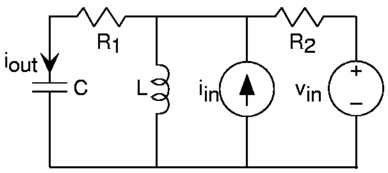
3.8 Using Impedances

Find the differential equation relating the indicated variable to the source(s) using impedances.

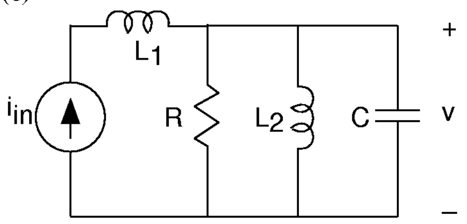
(a)


R_1
i_{out}
C
v_{in}
+
−
L
R_2

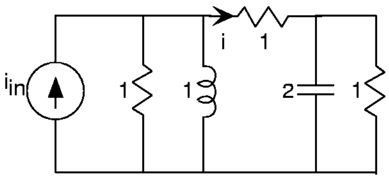
(b)


R_1
R_2
i_{out}
C
L
i_{in}
v_{in}
+
−

(c)

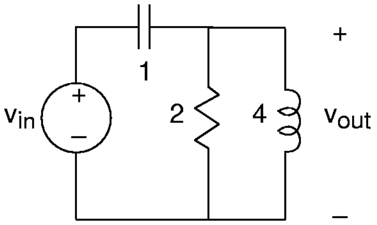

L_1
i_{in}
R
L_2
C
+
v
−

(d)



3.9 Transfer Functions

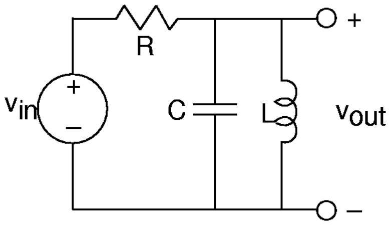
In the following circuit, the voltage source equals $v_{\text{in}}(t) = 10\sin(t/2)$.


1
v_{in}
+
−
2
4
+
v_{out}
−

(a) Find the transfer function between the source and the indicated output voltage.

(b) For the given source, find the output voltage.

3.10 Circuit Design



(a) Find the transfer function between the input and output voltages.

(b) At what frequency does the transfer function have a phase shift of zero? What is the circuit's gain at this frequency?

(c) Specifications demand that this circuit have an output impedance (its equivalent impedance) less than 8 Ω for frequencies above 1 kHz, the frequency at which the transfer function is maximum. Find element values that satisfy this criterion.

3.11 Circuit Detective Work

In the lab, the open-circuit voltage measured across an unknown circuit's terminals equals $\sin(t)$. When a 1 Ω resistor is placed across the terminals, a voltage of $\frac{1}{\sqrt{2}}\sin(t+\pi/4)$ appears.

(a) What is the Thévenin equivalent circuit?

(b) What voltage will appear if we place a 1 F capacitor across the terminals?

3.12 Linear, Time-Invariant Systems

For a system to be completely characterized by a transfer function, it needs not only to be linear, but also to be *time-invariant*. A system is said to be time-invariant if delaying the input delays the output by the same amount. Mathematically, if $\mathcal{S}[x(t)] = y(t)$, meaning $y(t)$ is the output of a system $\mathcal{S}[\cdot]$ when $x(t)$ is the input, $\mathcal{S}[\cdot]$ is time-invariant if $\mathcal{S}[x(t-\tau)] = y(t-\tau)$ for all delays τ and all inputs $x(t)$. Note that both linear and nonlinear systems have this property. For example, a system that squares its input is time-invariant.

(a) Show that if a circuit has fixed circuit elements (their values don't change over time), its input-output relationship is time-invariant. *Hint*: Consider the differential equation that describes a circuit's input-output relationship. What is its general form? Examine the derivative(s) of delayed signals.

(b) Show that impedances cannot characterize time-varying circuit elements (R, L, and C). Consequently, show that linear, time-varying systems do not have a transfer function.

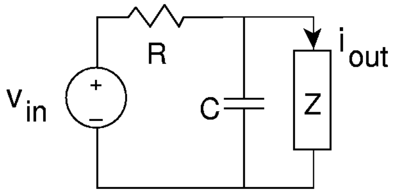
(c) Determine the linearity and time-invariance of the following. Find the transfer function of the linear, time-invariant (LTI) one(s).

(i) diode **(ii)** $y(t) = x(t)\cdot\sin 2\pi f_0 t$

(iii) $y(t) = x(t-\tau_0)$ **(iv)** $y(t) = x(t) + N(t)$

3.13 Long and Sleepless Nights

Sammy went to lab after a long, sleepless night, and constructed the following circuit.



He can't remember what the circuit, represented by the impedance Z, was. Clearly, this forgotten circuit is important as the output is the current passing through it.

(a) What is the Thévenin equivalent circuit seen by the impedance?

(b) In searching his notes, Sammy finds that the circuit is to realize the transfer function

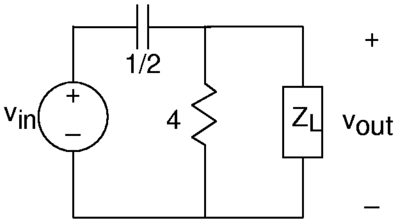
$$H(f) = \frac{1}{j10\pi f + 2}\ .$$

Find the impedance Z as well as values for the other circuit elements.

3.14 Find the Load Impedance

The depicted circuit has a transfer function between the output voltage and the source equal to

$$H(f) = \frac{-8\pi^2 f^2}{-8\pi^2 f^2 + 4 + j6\pi f}\ .$$



(a) Sketch the magnitude and phase of the transfer function.

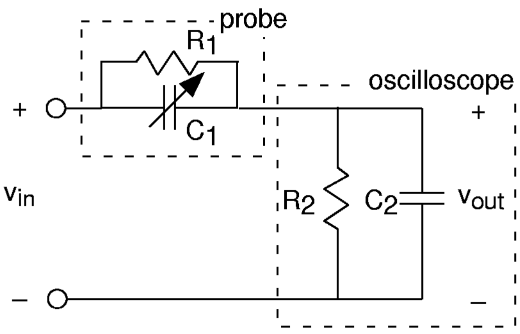
(b) At what frequency does the phase equal $\pi/2$?

(c) Find the load impedance Z_L that results in this transfer function.

(d) Find a circuit that corresponds to this load impedance. Is your answer unique? If so, show it to be so; if not, give another example.

3.15 An Interesting and Useful Circuit

The following circuit has interesting properties, which are exploited in high-performance oscilloscopes.



The portion of the circuit labeled oscilloscope represents the scope's input impedance. $R_2 = 1\ \mathrm{M}\Omega$ and $C_2 = 30$ pF (note the label under the channel 1 input in the lab's oscilloscopes). A *probe* is a device to attach an oscilloscope to a circuit, and has the indicated circuit inside it.

(a) Suppose for a moment that the probe is merely a wire and that the oscilloscope is attached to a circuit that has a resistive Thévenin equivalent impedance. What would be the effect of the oscilloscope's input impedance on measured voltages?

(b) Using the node method, find the transfer function relating the indicated voltage to the source when the probe is used.

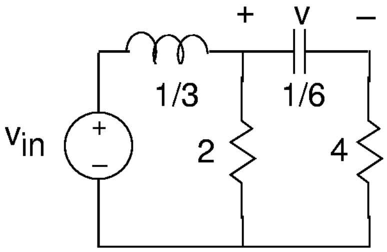
(c) Plot the magnitude and phase of this transfer function when $R_1 = 9\ \mathrm{M}\Omega$ and $C_1 = 2$ pF.

(d) For a particular relationship among the element values, the transfer function is quite simple. Find that relationship and describe what is so special about it.

(e) The arrow through C_1 indicates that its value can be varied. Select the value for this capacitor to make the special relationship valid. What is the impedance seen by the circuit being measured for this special value?

3.16 A Circuit Problem

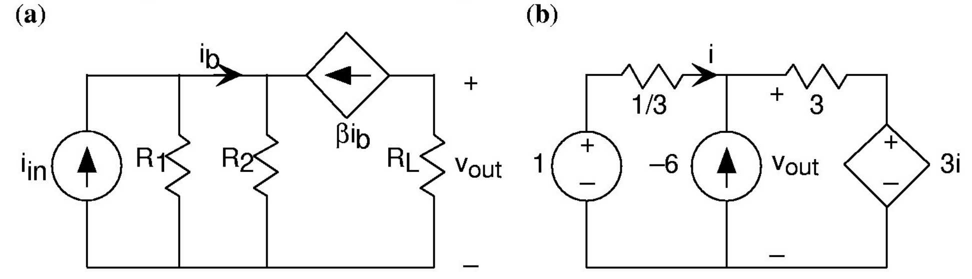
You are given the following circuit.



(a) Find the differential equation relating the output voltage to the source.

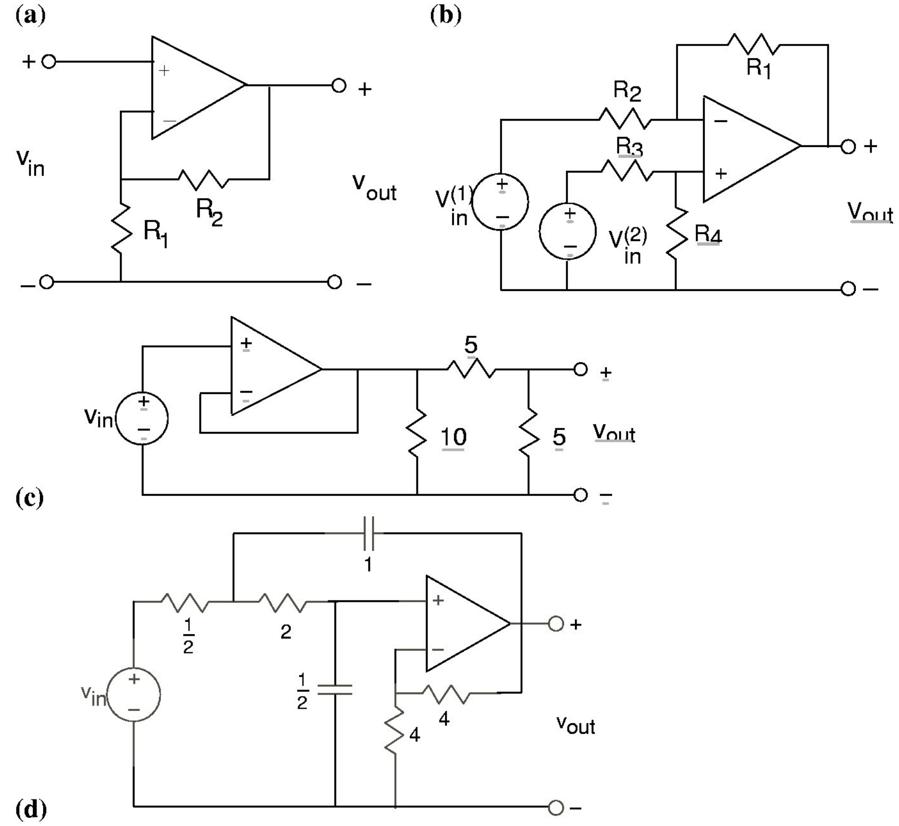
(b) What is the impedance "seen" by the capacitor?

3.17 Find the voltage v_{out} in each of the following circuits.


(a)
(b)
i_b
βi_b
i_{in}
R_1
R_2
R_L
+
v_{out}
–
i
1/3
3
1
–6
+
v_{out}
–
3i

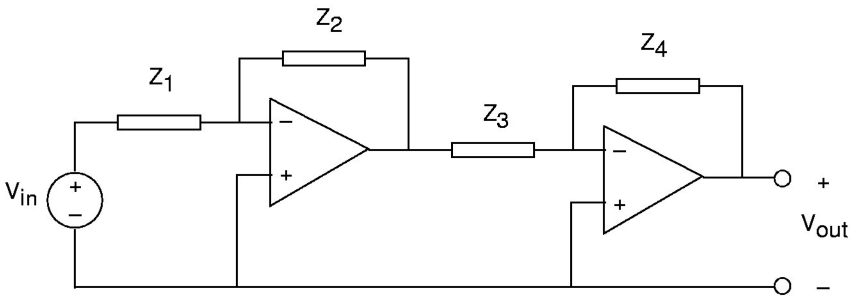
3.18 Operational Amplifiers

Find the transfer function between the source voltage(s) and the indicated output voltage.


(a)
(b)
(c)
(d)
+
–
v_{in}
v_{out}
R_1
R_2
R_3
R_4
$v_{in}^{(1)}$
$v_{in}^{(2)}$
v_{out}
v_{in}
5
10
5
v_{out}
1
1/2
2
1/2
4
4
v_{in}
v_{out}

3.19 Why Op-Amps are Useful

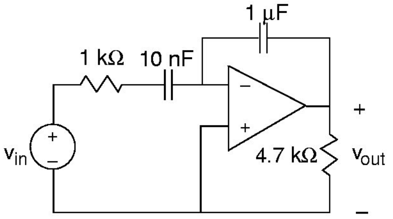
The following circuit of a cascade of op-amp circuits illustrates the reasons why op-amp realizations of transfer functions are so useful.



(a) Find the transfer function relating the complex amplitude of the voltage $v_{\text{out}}(t)$ to the source. Show that this transfer function equals the product of each stage's transfer function.

(b) What is the load impedance appearing across the first op-amp's output?

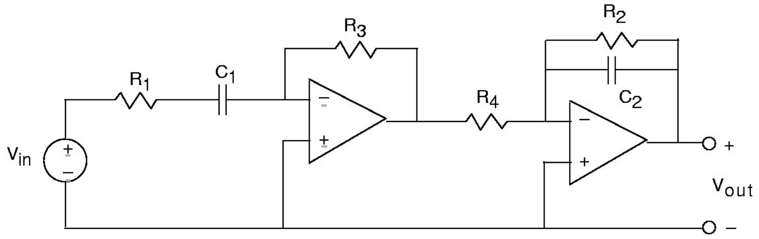
(c) The following illustrates that sometimes "designs" can go wrong.



Find the transfer function for this op-amp circuit, and then show that it can't work! Why can't it?

3.20 Operational Amplifiers

Consider the following circuit.

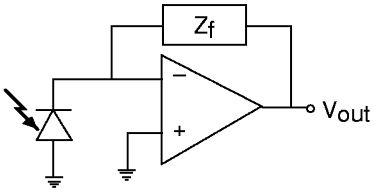


(a) Find the transfer function relating the voltage $v_{\text{out}}(t)$ to the source.

(b) In particular, $R_1 = 530\ \Omega$, $C_1 = 1\ \mu\text{F}$, $R_2 = 5.3\ \text{k}\Omega$, $C_2 = 0.01\ \mu\text{F}$, and $R_3, R_4 = 1\ \text{k}\Omega$. Characterize the resulting transfer function and determine what use this circuit might have.

3.21 Optical Receivers

In your optical telephone, the receiver circuit had the following form.

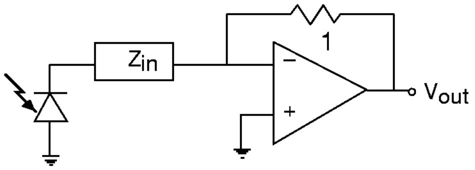


This circuit served as the transducer, converting light energy into a voltage v_{out}. The photodiode acts as a current source, producing a current proportional to the light intensity falling upon it. As is often the case in this crucial stage, the signals are small and noise can be a problem. Thus, the op-amp stage serves to boost the signal and to filter out-of-band noise.

(a) Find the transfer function relating light intensity to v_{out}.

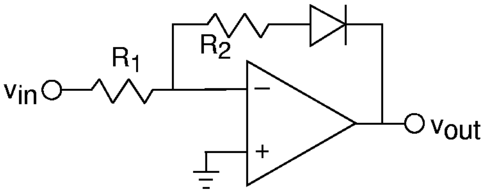
(b) What should the circuit realizing the feedback impedance Z_f be so that the transducer acts as a 5 kHz lowpass filter?

(c) A clever engineer suggests an alternative circuit to accomplish the same task. Determine whether the idea works or not. If it does, find the impedance Z_{in} that accomplishes the lowpass filtering task. If not, show why it does not work.



3.22 Rectifying Amplifier

The following circuit has been designed by a clever engineer. For positive voltages, it acts like a short circuit, and for negative voltages it acts like an open circuit.



(a) How is the output related to the input?

(b) Does this circuit have a transfer function? If so, find it; if not, why not?

Chapter 4

The Frequency Domain

In developing ways of analyzing linear circuits, we invented the impedance method because it made solving circuits easier. Along the way, we developed the notion of a circuit's frequency response or transfer function. This notion, which also applies to all linear, time-invariant systems, describes how the circuit responds to a sinusoidal input when we express it in terms of a complex exponential. We also learned the Superposition Principle for linear systems: The system's output to an input consisting of a sum of two signals is the sum of the system's outputs to each individual component.

This chapter combines these two notions —sinusoidal response and superposition—not only for systems, but also for signals, and develops the crucial idea of a signal's *spectrum.* Expressing the idea in a rudimentary way, all signals can be expressed as a superposition of sinusoids. As this chapter unfolds, we'll see that information systems rely heavily on spectral ideas. For example, radio, television, and cellular telephones transmit over different portions of the spectrum. In fact, spectrum is so important that what communications systems can use which portions of the spectrum is regulated by the Federal Communications Commission in the United States and by International Treaty for the world (see Appendix C). Calculating the spectrum is easy: This chapter develops the *Fourier transform* that defines how we can find a signal's spectrum.

4.1 Fourier Series

In a previous chapter, we found that we could express the square wave as a superposition of pulses. This superposition does not generalize well to other periodic signals, like a sinusoid. You might wonder whether a specific set of signals, when linearly combined, can represent virtually *any* signal. Pulses can't do it because they assume only constant values; signal's like sinusoids certainly aren't constant anywhere. Because of the importance of sinusoids to linear systems, you might wonder whether they could be used to represent periodic signals. You would be right and in good company as well. Euler and Gauss in particular worried about this problem, and Fourier got the credit even though tough mathematical issues were not settled until later. They worked on what is now known as the Fourier series.

Let $s(t)$ have period T. We want to show that periodic signals, even those that have constant-valued segments like a square wave, can be expressed as sum of harmonically related sine waves.

$$\boxed{s(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos\frac{2\pi kt}{T} + \sum_{k=1}^{\infty} b_k \sin\frac{2\pi kt}{T}} \tag{4.1}$$

The family of functions called *basis functions* $\{\cos 2\pi kt/T,\ \sin 2\pi kt/T\}$ form the foundation of the Fourier series. No matter what the periodic signal might be, these functions are always present and form the representation's building blocks. They do depend on the signal's period T, and are indexed by k. The frequency of each term is k/T. For $k=0$, the frequency is zero and the corresponding term a_0 is a constant. The basic frequency $1/T$ is called the *fundamental frequency* because all other terms have frequencies that are integer multiples of it. These higher frequency terms are called *harmonics*: The term at frequency $1/T$ is the fundamental and the first harmonic, $2/T$ the second harmonic, etc. Thus, larger values of the series index correspond to higher harmonic-frequency sinusoids. The *Fourier coefficients* a_k, b_k depend on the signal's waveform. Because frequency is linked to index, the coefficients implicitly depend on frequency. *Assuming we know the period, knowing the Fourier coefficients is equivalent to knowing the signal.*

Assume for the moment that the Fourier series works. To find the Fourier coefficients, we note the following *orthogonality* properties of sinusoids.

$$\begin{aligned}
&\int_0^T \sin\frac{2\pi kt}{T}\cos\frac{2\pi lt}{T}\,dt = 0 \text{ for all } k, l\\
&\int_0^T \sin\frac{2\pi kt}{T}\sin\frac{2\pi lt}{T}\,dt = \begin{cases}\frac{T}{2}, & k=l; k,l\neq 0\\ 0, & k\neq l; k=0=l\end{cases}\\
&\int_0^T \cos\frac{2\pi kt}{T}\cos\frac{2\pi lt}{T}\,dt = \begin{cases}\frac{T}{2}, & k=l; k,l\neq 0\\ T, & k=0=l\\ 0, & k\neq l\end{cases}
\end{aligned} \tag{4.2}$$

To use these, let's multiply the Fourier series for $s(t)$ by $\cos 2\pi lt/T$ and integrate. The idea is that, because integration is linear, the integration will sift out all but the term involving a_l.

$$\begin{aligned}\int_0^T s(t)\cos\frac{2\pi lt}{T}\,dt = &\int_0^T a_0\cos\frac{2\pi lt}{T}\,dt\\ &+\sum_{k=1}^{\infty} a_k\int_0^T \cos\frac{2\pi kt}{T}\cos\frac{2\pi lt}{T}\,dt\\ &+\sum_{k=1}^{\infty} b_k\int_0^T \sin\frac{2\pi kt}{T}\cos\frac{2\pi lt}{T}\,dt\end{aligned}$$

The first and third terms are zero; in the second, the only non-zero term in the sum results when the indices k and l are equal (but not zero), in which case we obtain $a_lT/2$. If $k=0=l$, we obtain a_0T. Consequently,

$$a_l = \frac{2}{T}\int_0^T s(t)\cos\frac{2\pi lt}{T}\,dt,\quad l\neq 0\ .$$

All of the Fourier coefficients can be found similarly.

$$\boxed{\begin{aligned}a_0 &= \frac{1}{T}\int_0^T s(t)\,dt\\ a_k &= \frac{2}{T}\int_0^T s(t)\cos\frac{2\pi kt}{T}\,dt,\quad k\neq 0\\ b_k &= \frac{2}{T}\int_0^T s(t)\sin\frac{2\pi kt}{T}\,dt\end{aligned}} \tag{4.3}$$

Question 4.1 {250} *The expression for a_0 is referred to as the* average value *of $s(t)$. Why?*

Example

Let's find the Fourier series representation for the half-wave rectified sinusoid.

$$s(t) = \begin{cases}\sin\frac{2\pi t}{T}, & 0\leq t<\frac{T}{2}\\ 0, & \frac{T}{2}\leq t<T\end{cases}$$

Begin with the sine terms in the series; to find b_k, we must calculate the integral

$$b_k = \frac{2}{T}\int_0^{T/2}\sin\frac{2\pi t}{T}\sin\frac{2\pi kt}{T}\,dt\ .$$

The key to evaluating such integrals is the classic trigonometric identities.

$$\boxed{\begin{aligned}\sin\alpha\sin\beta &= \frac{1}{2}\left(\cos(\alpha-\beta)-\cos(\alpha+\beta)\right)\\ \cos\alpha\cos\beta &= \frac{1}{2}\left(\cos(\alpha+\beta)+\cos(\alpha-\beta)\right)\\ \sin\alpha\cos\beta &= \frac{1}{2}\left(\sin(\alpha+\beta)+\sin(\alpha-\beta)\right)\end{aligned}} \tag{4.4}$$

Using these turns our integral of a product of sinusoids into a sum of integrals of individual sinusoids, which are much easier to evaluate.

$$\begin{aligned}\int_0^{T/2}\sin\frac{2\pi t}{T}\sin\frac{2\pi kt}{T}\,dt &= \frac{1}{2}\int_0^{T/2}\left[\cos\frac{2\pi(k-1)t}{T}-\cos\frac{2\pi(k+1)t}{T}\right]dt\\ &= \begin{cases}\frac{1}{2}, & k=1\\ 0, & k=2,3,\dots\end{cases}\end{aligned}$$

Thus, $b_1=\frac{1}{2}$, $b_2=b_3=\cdots=0$.

On to the cosine terms. The average value, which corresponds to a_0, equals $\frac{1}{\pi}$. The remainder of the cosine coefficients are easy to find, but yield the complicated result

$$a_k=\begin{cases}-\frac{2}{\pi}\frac{1}{k^2-1}, & k=2,4,\dots\\ 0, & k\text{ odd}\end{cases}.$$

Thus, the Fourier series for the half-wave rectified sinusoid has non-zero terms for the average, the fundamental, and the even harmonics. Plotting the Fourier coefficients reveals at what component frequencies the half-wave rectified sinusoid has energy (Figure 4.1 {87}). Furthermore, this figure shows what the Fourier series sum looks like with these coefficients as we add more and more terms. Presumably, you now believe more in the Fourier series.

4.1.1 A Signal's Spectrum

Thus, a periodic signal, such as the half-wave rectified sinusoid, consists of a sum of elemental sinusoids. A plot of the Fourier coefficients as a function of the frequency index, such as shown in Figure 4.1 {87}, displays the signal's *spectrum*. The word "spectrum" implies that the independent variable, here k, corresponds somehow to

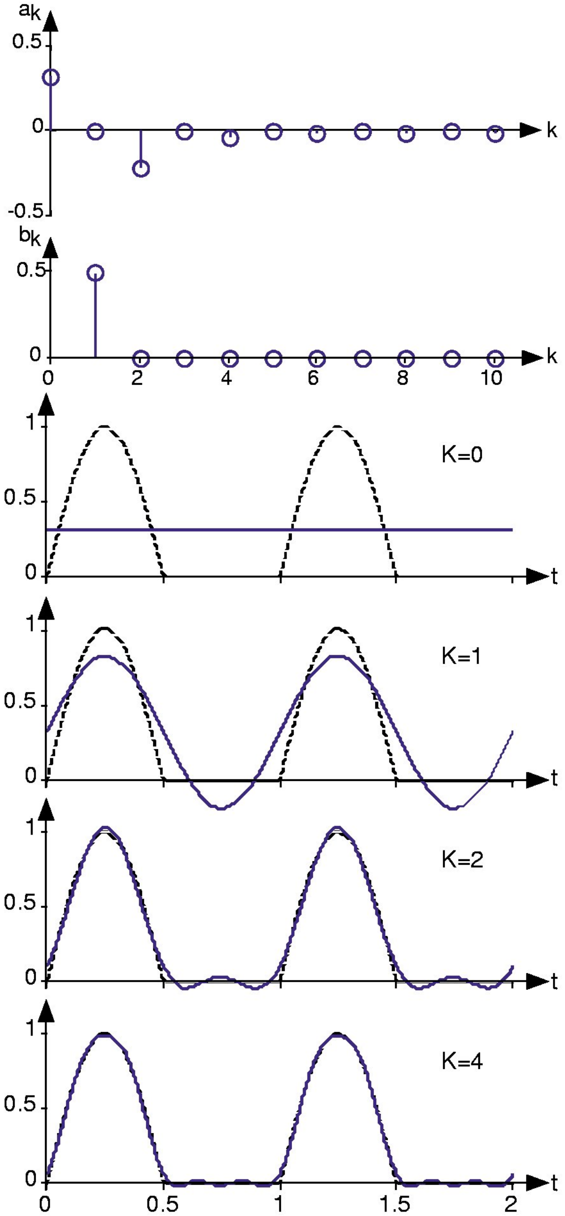


Figure 4.1 The Fourier series spectrum of a half-wave rectified sinusoid is shown in the upper portion. The index indicates the multiple of the fundamental frequency at which the signal has energy. The cumulative effect of adding terms to the Fourier series for the half-wave rectified sine wave is shown in the bottom portion. The dashed line is the actual signal, with the solid line showing the finite series approximation to the indicated number of terms K.

frequency. Each coefficient is directly related to a sinusoid having a frequency of k/T. Thus, if we half-wave rectified a 1 kHz sinusoid, $k = 1$ corresponds to 1 kHz, $k = 2$ to 2 kHz, etc. We do indeed have a spectrum.

A subtle, but very important, aspect of the Fourier spectrum is its *uniqueness*: You can unambiguously find the spectrum from the signal (Eq. (4.3) $\{85\}$) and the signal from the spectrum (Eq. (4.1) $\{84\}$). Thus, any aspect of the signal can be found from the spectrum and vice versa. *A signal's frequency domain expression is its spectrum.* A periodic signal can be defined either in the time domain (as a function) or in the frequency domain (as a Fourier series through the Fourier coefficients). A fundamental aspect of solving electrical engineering problems is whether the time or frequency domain provides the most understanding and the simplest way of manipulating signals and understanding their properties. The uniqueness property says that either domain can provide the right answer. As a simple example, suppose we want to know the (periodic) signal's maximum value. Clearly the time domain provides the answer directly. To use a frequency domain approach would require us to find the spectrum, form the signal from the spectrum, and calculate the maximum: We're back in the time domain!

A more interesting question you could ask about a signal is its average *power*. A signal's instantaneous power is defined to be its square. The average power is the average of the instantaneous power over some time interval. For a periodic signal, the natural time interval is clearly its period; for nonperiodic signals, a better choice would be entire time or time from onset. For a periodic signal, the average power is the square of the root-mean-squared (rms) value. We define the rms value of a periodic signal to be (see Problem 1.1 $\{9\}$)

$$\mathrm{rms}[s] = \sqrt{\frac{1}{T}\int_0^T s^2(t)\,dt}$$

and thus its average power is $\mathrm{rms}^2[s]$.

$$\mathrm{power}[s] = \mathrm{rms}^2[s] = \frac{1}{T}\int_0^T s^2(t)\,dt$$

Question 4.2 $\{$**250**$\}$ *What is the rms value of the half-wave rectified sinusoid? If you worked Problem 1.1* $\{9\}$*, think a little before answering. You should be able to determine the answer here from that one.*

To find the average power in the frequency domain, we need to substitute the spectral representation of the signal into this expression.

$$\mathrm{power}[s] = \frac{1}{T}\int_0^T \left[a_0 + \sum_{k=1}^{\infty} a_k \cos\frac{2\pi kt}{T} + \sum_{k=1}^{\infty} b_k \sin\frac{2\pi kt}{T}\right]^2 dt$$

The square inside the integral will contain all possible pairwise products. However, the orthogonality properties (4.2) {84} say that most of these crossterms integrate to zero. The survivors leave a rather simple expression for the power we seek.

$$\boxed{\text{power}[s] = a_0^2 + \frac{1}{2}\sum_{k=1}^{\infty}\left(a_k^2 + b_k^2\right)}$$

It could well be that computing this sum is easier than integrating the signal's square. Furthermore, the contribution of each term in the Fourier series toward representing the signal can be measured by its contribution to the signal's average power. Thus, the power contained in a signal at its k^{th} harmonic is $(a_k^2 + b_k^2)/2$. The *power spectrum* $\mathcal{P}_s(k)$, such as shown in Figure 4.2, plots each harmonic's contribution to the total power.

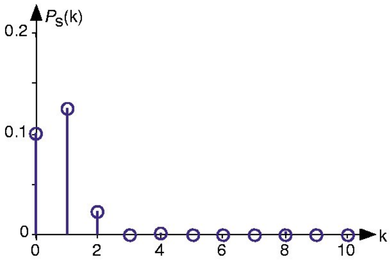


Figure 4.2 Power spectrum of a half-wave rectified sinusoid.

Question 4.3 {250} *In stereophonic systems, deviation of a sine wave from the ideal is measured by the* total harmonic distortion, *which equals the total power in the harmonics higher than the first compared to power in the fundamental. Find an expression for the total harmonic distortion for any periodic signal. Is this calculation most easily performed in the time or frequency domain?*

It is interesting to consider the sequence of signals that obtain as we incorporate more terms to the series. Define $s_K(t)$ to be the signal containing K Fourier terms.

$$s_K(t) = a_0 + \sum_{k=1}^{K} a_k \cos\frac{2\pi kt}{T} + \sum_{k=1}^{K} b_k \sin\frac{2\pi kt}{T}$$

Figure 4.1 shows how this sequence of signals increasingly portrays the signal accurately as more terms are added. We need to assess quantitatively the accuracy of the N-term Fourier series approximation so that we can judge how rapidly the series approaches the signal. When we use an $K+1$-term series, the error —the difference between the signal and the $K+1$-term series —corresponds to the unused terms from the series.

$$\epsilon_K(t) = \sum_{k=K+1}^{\infty} a_k \cos\frac{2\pi kt}{T} + \sum_{k=K+1}^{\infty} b_k \sin\frac{2\pi kt}{T}$$

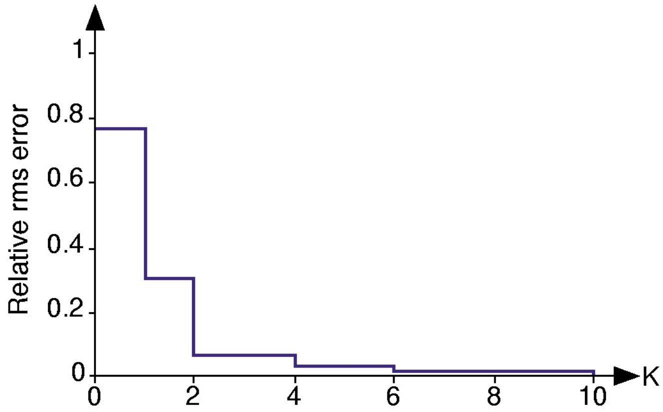


Figure 4.3 The rms error calculated according to (4.5) is shown as a function of the number of terms in the series. The error has been normalized by the rms value of the signal.

To find the rms error, we must square this expression and integrate it over a period. Again, the integral of most cross-terms is zero, leaving

$$\mathrm{rms}[\epsilon_K] = \sqrt{\frac{1}{2}\sum_{k=K+1}^{\infty}\left(a_k^2 + b_k^2\right)}\ . \tag{4.5}$$

Figure 4.3 shows how the error in the Fourier series decreases as more terms are incorporated. In particular, the use of four terms, as shown in the bottom plot of Figure 4.1, has a rms error (relative to the rms value of the signal) of about 3%. The Fourier series in this case converges quickly to the signal.

4.1.2 Equality of Representations

The Fourier series representation of a signal, as expressed by (4.1) $\{84\}$ says that the left and right sides are "equal." We need to explore, through an example, what equality means.

Example

Let's find the spectrum of the square wave $\mathrm{sq}(t)$. The expressions for the Fourier coefficients have the common form

$$\left\{\begin{matrix} a_k \\ b_k \end{matrix}\right\} = \frac{1}{T}\int_0^{T/2}\left\{\begin{matrix}\cos\frac{2\pi kt}{T} \\ \sin\frac{2\pi kt}{T}\end{matrix}\right\} dt - \frac{1}{T}\int_{T/2}^{T}\left\{\begin{matrix}\cos\frac{2\pi kt}{T} \\ \sin\frac{2\pi kt}{T}\end{matrix}\right\} dt$$

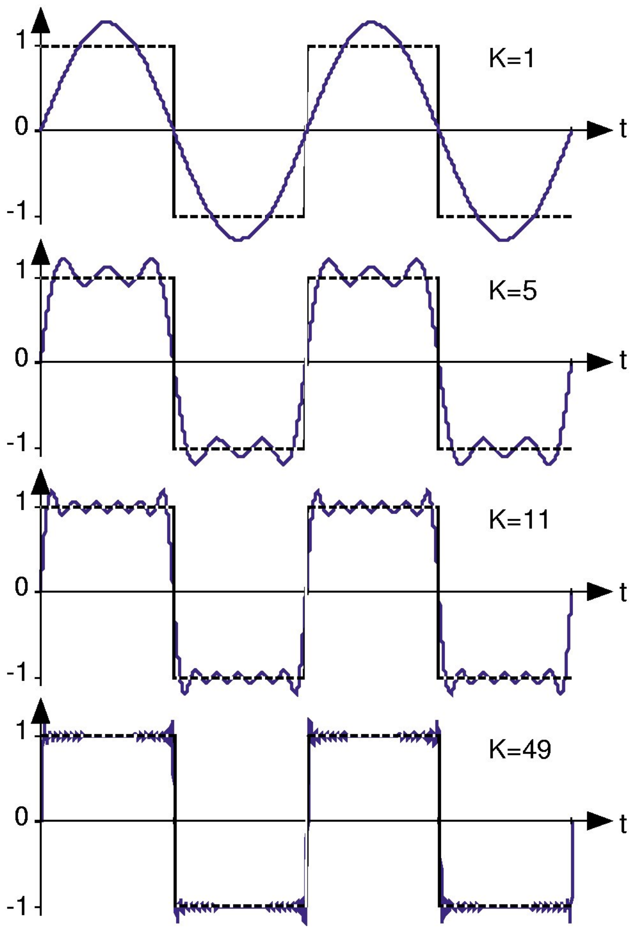


Figure 4.4 Fourier series approximation to $\mathrm{sq}(t)$. The number of terms in the Fourier sum is indicated in each plot, and the square wave is shown as a dashed line over two periods.

The cosine coefficients a_k are all zero and the sine coefficients are

$$b_k = \begin{cases} \dfrac{4}{\pi k}, & k \text{ odd} \\ 0, & k \text{ even} \end{cases}$$

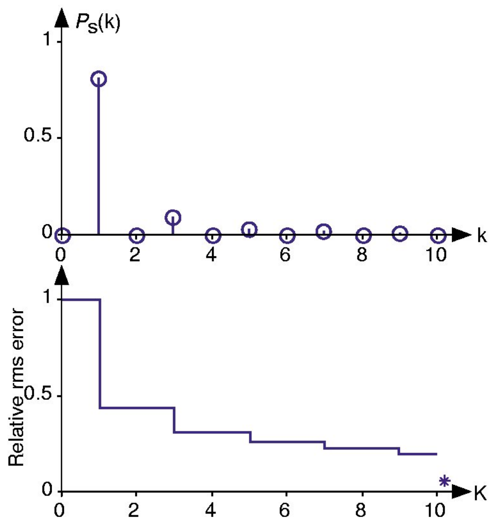


Figure 4.5 The upper plot shows the power spectrum of the square wave, and the lower plot the rms error of the finite-length Fourier series approximation to the square wave. The asterisk denotes the rms error when the number of terms K in the Fourier series equals 99.

Thus, the Fourier series for the square wave is

$$\boxed{\mathrm{sq}(t) = \sum_{k=1,3,\ldots} \frac{4}{\pi k} \sin \frac{2\pi kt}{T}} \tag{4.6}$$

As we see in Figure 4.4 {91}, the Fourier series requires many more terms to provide the same quality of approximation as we found with the half-wave rectified sinusoid (Figure 4.1 {87}). We can verify that more terms are needed by considering the power spectrum and the approximation error shown in Figure 4.5. This difference between the two Fourier series results because the half-wave rectified sinusoid's Fourier coefficients are proportional to $1/k^2$ while those of the square wave are proportional to $1/k$. In short, the square wave's coefficients decay more slowly with increasing frequency. Said another way, the square-wave's spectrum contains more power at higher frequencies than does the half-wave-rectified sinusoid.

Question 4.4 {250} *Calculate the harmonic distortion for the square wave.*

The fact that the square wave's Fourier series requires more terms for a given representation accuracy is not important. Close inspection of Figure 4.4 {91} does reveal a potential issue: Does the Fourier series really equal the square wave at *all* values of t? In particular, at each step-change in the square wave, the Fourier series exhibits a peak followed by rapid oscillations. As more terms are added to the series, the oscillations seem to become more rapid and smaller, but the peaks are not decreasing. Consider this mathematical question intuitively: Can a discontinuous function, like the square wave, be expressed as a sum, even an infinite one, of continuous ones? One should at least be suspicious, and in fact, it can't be thus expressed. This issue brought Fourier much criticism from the French Academy of Science (Laplace, Legendre, and Lagrange comprised the review committee) for several years after its presentation on 1807. It was not resolved for also a century, and its resolution is interesting and important to understand from a practical viewpoint.

The extraneous peaks in the square wave's Fourier series *never* disappear; they are termed *Gibb's phenomenon* after the American physicist Josiah Willard Gibbs. They occur whenever the signal is discontinuous, and will always be present whenever the signal has jumps. Let's return to the question of equality; how can the equal sign in the definition of the Fourier series ((4.1) $\{84\}$) be justified? The partial answer is that pointwise —each and every value of t —equality is *not* guaranteed. What mathematicians later in the nineteenth century showed was that the rms error of the Fourier series was always zero.

$$\lim_{K\to\infty} \mathrm{rms}[\epsilon_K] = 0$$

What this means is that the difference between an actual signal and its Fourier series representation may not be zero, but the square of this quantity has *zero* integral! It is through the eyes of the rms value that we define equality: Two signals $s_1(t)$, $s_2(t)$ are said to be equal in the *mean square* if $\mathrm{rms}[s_1 - s_2] = 0$. These signals are said to be equal *pointwise* if $s_1(t) = s_2(t)$ for all values of t. For Fourier series, Gibb's phenomenon peaks have finite height and zero width: The error differs from zero only at isolated points —whenever the periodic signal contains discontinuities—and equals about 9% of the size of the discontinuity. The value of a function at a finite set of points does not affect its integral. This effect underlies the reason why defining the value of a discontinuous function, like we refrained from doing in defining the step function $\{14\}$, at its discontinuity is meaningless. Whatever you pick for a value has no practical relevance for either the signal's spectrum or for how a system responds to the signal. The Fourier series value "at" the discontinuity is the average the values on either side of the jump.

4.2 Complex Fourier Series

We can greatly simplify the expressions for a signal's Fourier series by using Euler's relations (Eq. (2.1) {12}). In this way, we have that when we combine cosine and sine terms at the same frequency

$$a_k \cos\frac{2\pi kt}{T} + b_k \sin\frac{2\pi kt}{T} = \frac{1}{2}\left[a_k e^{j\frac{2\pi kt}{T}} - jb_k e^{j\frac{2\pi kt}{T}}\right] + \frac{1}{2}\left[a_k e^{-j\frac{2\pi kt}{T}} + jb_k e^{-j\frac{2\pi kt}{T}}\right] .$$

By defining $c_k = \frac{1}{2}(a_k - jb_k), k \neq 0$ and $c_0 = a_0$, we can rewrite the Fourier series of (4.1) {84} as

$$\boxed{s(t) = \sum_{k=-\infty}^{\infty} c_k e^{j\frac{2\pi kt}{T}}} \tag{4.7}$$

with $c_{-k} = c_k^* = \frac{1}{2}(a_k + jb_k), k \neq 0$. In this way, we have the *complex Fourier series*. The positive index terms correspond to the first portion of the common-frequency term and the negative indexed terms to the second. Thus, *the complex Fourier series and the sine-cosine series are identical*, each representing a signal's spectrum in slightly different ways. Manipulations and calculations are more streamlined with complex Fourier series, and it is our representation of choice.

To find the Fourier coefficients, we note the orthogonality property

$$\int_0^T e^{j\frac{2\pi kt}{T}} e^{-j\frac{2\pi lt}{T}}\, dt = \begin{cases} T & k = l \\ 0 & k \neq l \end{cases}$$

We can find a signal's complex Fourier spectrum with

$$\boxed{c_k = \frac{1}{T}\int_0^T s(t) e^{-j\frac{2\pi kt}{T}}\, dt} \tag{4.8}$$

The complex Fourier series for the square wave is

$$\mathrm{sq}(t) = \sum_{k=\ldots,-3,-1,1,3,\ldots} \frac{2}{j\pi k} e^{+j\frac{2\pi kt}{T}} .$$

A signal's Fourier series spectrum c_k has interesting properties.

- If $s(t)$ is real, $c_{-k} = c_k^*$.

  This result follows from the integral that calculates the c_k from the signal. Furthermore, this result means that $\mathrm{Re}\,[c_k] = \mathrm{Re}\,[c_{-k}]$: The real part of the Fourier coefficients for real-valued signals is even. Similarly, $\mathrm{Im}\,[c_k] = -\mathrm{Im}\,[c_{-k}]$: The imaginary parts of the Fourier coefficients have odd symmetry. Consequently, if you are given the Fourier coefficients for positive indices and zero and are told the signal is real-valued, you can find the negative-indexed coefficients, hence the entire spectrum. This kind of symmetry—$c_{-k} = c_k^*$—is known as *conjugate symmetry.* We can phrase the property concisely by saying *real-valued periodic signals have a conjugate-symmetric spectrum.*

- If $s(-t) = s(t)$, which says the signal has even symmetry about the origin, $c_{-k} = c_k$.

  Given the previous property for real-valued signals, the Fourier coefficients of even signals are real-valued. A real-valued Fourier expansion amounts to an expansion in terms of only cosines, which is the simplest example of an even signal.

- If $s(-t) = -s(t)$, which says the signal has odd symmetry, $c_{-k} = -c_k$.

  Therefore, the Fourier coefficients are purely imaginary. The square wave is a great example of an odd-symmetric signal.

- The spectral coefficients for the periodic signal $s(t - \tau)$ are $c_k e^{-j2\pi k\tau/T}$, where c_k denotes the spectrum of $s(t)$.

  Thus, delaying a signal by τ seconds results in a spectrum having a *linear phase shift* of $-2\pi k\tau/T$ in comparison to the spectrum of the undelayed signal. Note that the spectral magnitude is unaffected. Showing this property is easy.

$$\begin{aligned}\frac{1}{T}\int_0^T s(t-\tau)e^{-j\frac{2\pi kt}{T}}\,dt &= \frac{1}{T}\int_{-\tau}^{T-\tau} s(t)e^{-j\frac{2\pi k(t+\tau)}{T}}\,dt\\ &= \frac{1}{T}e^{-j\frac{2\pi k\tau}{T}}\int_{-\tau}^{T-\tau} s(t)e^{-j\frac{2\pi kt}{T}}\,dt\end{aligned}$$

  At this point, the range of integration extends over a period of the integrand. Consequently, it should not matter how we integrate over a period, which means that $\int_{-\tau}^{T-\tau} = \int_0^T$, and we have our result.

- The Fourier series obeys *Parseval's Theorem*: Power calculated in the time domain equals the power calculated in the frequency domain.

$$\boxed{\frac{1}{T}\int_0^T s^2(t)\,dt = \sum_{k=-\infty}^{\infty} |c_k|^2} \tag{4.9}$$

This result is a (simpler) re-expression of how to calculate a signal's power with the real-valued Fourier series {89}.

Example

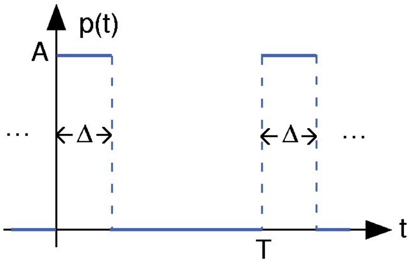


Figure 4.6 Periodic pulse signal.

Let's calculate the spectrum of the periodic pulse signal shown here. The pulse width is Δ, the period T, and the amplitude A. The complex Fourier spectrum of this signal is given by

$$\begin{aligned} c_k &= \frac{A}{T}\int_0^{\Delta} e^{-j\frac{2\pi k t}{T}}\,dt \\ &= -\frac{A}{j2\pi k}\left[e^{-j\frac{2\pi k\Delta}{T}} - 1\right] \\ c_k &= Ae^{-j\frac{\pi k\Delta}{T}}\frac{\sin\frac{\pi k\Delta}{T}}{\pi k} \end{aligned} \tag{4.10}$$

Because this signal is real-valued, we find that the coefficients do indeed have conjugate symmetry. The periodic pulse signal has neither even nor odd symmetry; consequently, no additional symmetry exists in the spectrum. Because the spectrum is complex valued, to plot it we need to calculate its magnitude and phase.

$$|c_k| = A\left|\frac{\sin\frac{\pi k\Delta}{T}}{\pi k}\right| \tag{4.11a}$$

$$\angle c_k = -\frac{\pi k\Delta}{T} + \pi\,\mathrm{neg}\left(\frac{\sin\frac{\pi k\Delta}{T}}{\pi k}\right) \tag{4.11b}$$

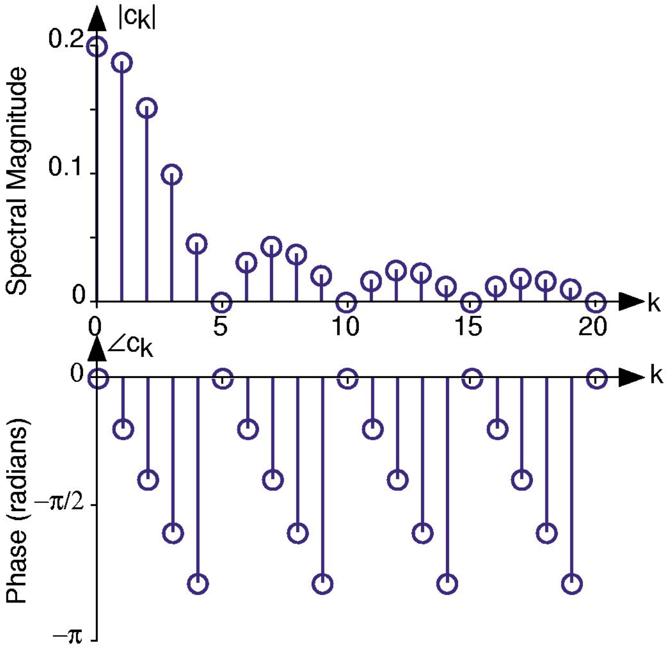


Figure 4.7 The magnitude and phase of the periodic pulse sequence's spectrum is shown for positive-frequency indices. Here $\Delta/T = 0.2$ and $A = 1$.

Question 4.5 {**251**} *What is the value of* c_0*? Recalling that this spectral coefficient corresponds to the signal's average value, does your answer make sense?*

The somewhat complicated expression for the phase results because the sine term can be negative; magnitudes must be positive, leaving the occasional negative values to be accounted for as a phase shift of π. The neg$(\cdot)$ function equals -1 if its argument is negative and zero otherwise. Also note the presence of a linear phase term (the first term in $\angle c_k$ is proportional to frequency k/T). Comparing this term with that predicted from delaying a signal, a delay of $\Delta/2$ is present in our signal. Advancing the signal by this amount centers the pulse about the origin, leaving an even signal, which in turn means that its spectrum is real-valued. Thus, our calculated spectrum is consistent with the properties of the Fourier spectrum.

Question 4.6 {**251**} *Investigate the half-wave rectified sine wave's spectrum for a linear phase term. If one is present, show how to delay or advance the signal to create an even or odd signal. If one is not present, convince yourself that no delay* *would yield a signal having even or odd symmetry.*

The phase plot shown in Fig. 4.7 requires some explanation as it does not seem to agree with what Eq. 4.11b suggests. There, the phase has a linear component, with a jump of π every time the sinusoidal term changes sign. We must realize that any integer multiple of 2π can be added to a phase at each frequency *without* affecting the value of the complex spectrum. We see that at frequency index 4 the phase is nearly $-\pi$. The phase at index 5 is undefined because the magnitude is zero in this example. At index 6, the formula suggests that the phase of the linear term should be less than (more negative) than $-\pi$. In addition, we expect a shift of $-\pi$ in the phase between indices 4 and 6. Thus, the phase value predicted by the formula is a little less than -2π. Because we can add 2π without affecting the value of the spectrum at index 6, the result is a slightly negative number as shown. Thus, the formula and the plot do agree. In phase calculations like those made in MATLAB, values are usually confined to the range $[-\pi, \pi)$ by adding some (possibly negative) multiple of 2π to each phase value.

4.3 Encoding Information in the Frequency Domain

To emphasize the fact that every periodic signal has both a time and frequency domain representation, we can exploit both to *encode information* into a signal. Refer to the Fundamental Model of Communication shown in Figure 1.1 $\{4\}$. We have an information source, and want to construct a transmitter that produces a signal $x(t)$. For the source, let's assume we have information to encode every T seconds. For example, we want to represent typed letters produced by an extremely good typist (a key is struck every T). Let's consider the complex Fourier series formula in the light of trying to encode information.

$$x(t) = \sum_{k=-K}^{K} c_k e^{j\frac{2\pi kt}{T}}$$

We use a finite sum here merely for simplicity (fewer parameters to determine). An important aspect of the spectrum is that each frequency component c_k can be manipulated separately: Instead of finding the Fourier spectrum from a time-domain specification, let's construct it in the frequency domain by selecting the c_k according to some rule that relates coefficient values to the alphabet. In defining this rule, we want to always create a real-valued signal $x(t)$. Because of the Fourier spectrum's properties $\{95\}$, the spectrum must therefore have conjugate symmetry.

This requirement means that we can only assign positive-indexed coefficients (positive frequencies), with negative-indexed ones equaling the complex conjugate of the corresponding positive-indexed ones.

Assume we have N letters to encode: $\ell_1, \dots, \ell_N$. One simple encoding rule could be to make a single Fourier coefficient be non-zero and all others zero for each letter. For example, if ℓ_n occurs, we make $c_n = 1$ and $c_k = 0, k \neq n$. In this way, the n^{th} harmonic of the frequency $1/T$ is used to represent a letter. Note that the *bandwidth*—the range of frequencies required for the encoding —equals N/T. Another possibility is to consider the binary representation of the letter's index. For example, if the letter ℓ_{13} occurs, converting 13 to its base 2 representation, we have $13_{10} = 1101_2$. We can use the pattern of zeros and ones to represent directly which Fourier coefficients we "turn on" (set equal to one) and which we "turn off".

Question 4.7 $\{251\}$ *Compare the bandwidth required for the direct encoding scheme (one nonzero Fourier coefficient for each letter) to the binary number scheme. Compare the bandwidths for a 128-letter alphabet. Since both schemes represent information without loss —we can determine the typed letter uniquely from the signal's spectrum —both are viable. Which makes more efficient use of bandwidth and thus might be preferred?*

Question 4.8 $\{251\}$ *Can you think of an information-encoding scheme that makes even more efficient use of the spectrum? In particular, can we use only one Fourier coefficient to represent N letters uniquely?*

As this information-encoding scheme now stands, we can represent one letter for all time. However, we note that the Fourier coefficients depend *only* on the signal's characteristics over a single period. We could change the signal's spectrum every T as each letter is typed. In this way, we turn spectral coefficients on and off as letters are typed, thereby encoding the entire typed document. For the receiver (Fig. 1.1) to retrieve the typed letter, it would simply use the Fourier formula of Eq. (4.8) $\{94\}$ for each T-second interval to determine what each typed letter was. Fig. 4.8 shows such a signal in the time-domain.

In this Fourier-series encoding scheme, we have used the fact that spectral coefficients can be independently specified and that they can be uniquely recovered from the time-domain signal over one "period." Do note that the signal representing the entire document is no longer periodic. By understanding the Fourier series' properties (in particular that coefficients are determined only over a T-second interval, we can construct a communications system. This approach represents a simplification of how modern modems represent text so that they can be transmitted over telephone lines.

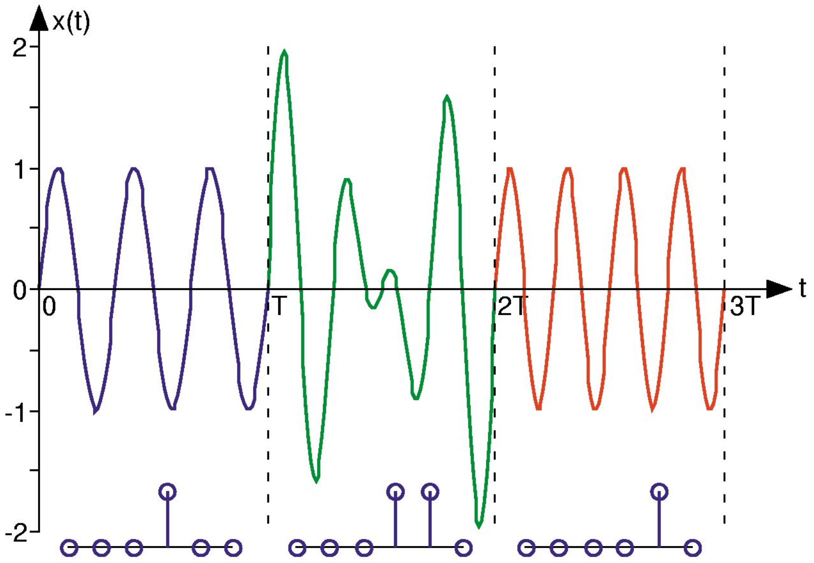


Figure 4.8 The encoding of signals via the Fourier spectrum is shown over three "periods." In this example, only the third and fourth harmonics are used, as shown by the spectral magnitudes corresponding to each T-second interval plotted below the waveforms. Can you determine the phase of the harmonics from the waveform?

4.4 Filtering Periodic Signals

The Fourier series representation of a periodic signal makes it easy to determine how a linear, time-invariant filter reshapes such signals *in general*. The fundamental property of a linear system is that its input-output relation obeys superposition: $\mathcal{L}\left[a_1 s_1(t) + a_2 s_2(t)\right] = a_1\,\mathcal{L}\left[s_1(t)\right] + a_2\,\mathcal{L}\left[s_2(t)\right]$. Because the Fourier series represents a periodic signal as a linear combination of complex exponentials, we can exploit the superposition property. Furthermore, we found for linear circuits that their output to a complex exponential input is just the frequency response evaluated at the signal's frequency times the complex exponential. Said mathematically, if $x(t) = e^{j\frac{2\pi kt}{T}}$, then the output $y(t) = H\left(\frac{k}{T}\right) e^{j\frac{2\pi kt}{T}}$. Thus, if $x(t)$ is periodic, a linear circuit's output to this signal is

$$y(t) = \sum_{k=-\infty}^{\infty} c_k H\left(\frac{k}{T}\right) e^{j\frac{2\pi kt}{T}} \,. \tag{4.12}$$

Thus, the output has a Fourier series, which means that it too is periodic. Its Fourier coefficients equal $c_k H\left(\frac{k}{T}\right)$. *To obtain the spectrum of the output, we simply multi-*

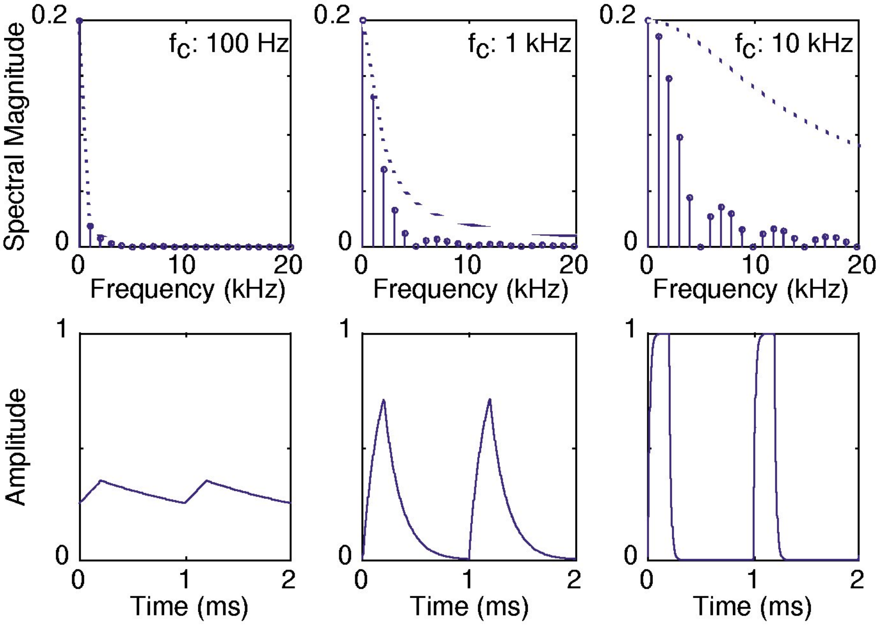


Figure 4.9 A periodic pulse signal, such as shown in Figure 4.6 {96} ($\Delta/T = 0.2$), serves as the input to an RC lowpass filter. The input's period was 1 ms (millisecond). The filter's cutoff frequency was set to the various indicated in the top row, which display the output signal's spectrum and the filter's transfer function. The bottom row shows the output signal derived from the Fourier series coefficients shown in the top row.

ply the input spectrum by the frequency response. The circuit modifies the magnitude and phase of each Fourier coefficient. Note especially that while the Fourier coefficients do not depend on the signal's period, the circuit's transfer function does depend on frequency, which means that the circuit's output will differ according to the period.

Example

The periodic pulse signal used in the previous example {96} serves as the input to a RC-circuit that has the transfer function {43}

$$H(f) = \frac{1}{1 + j2\pi fRC} .$$

Figure 4.9 shows the output changes as we vary the filter's cutoff frequency. Note how the signal's spectrum extends well above its fundamental frequency. Having a cutoff frequency ten times higher than the fundamental does perceptibly change the input waveform, rounding the leading and trailing edges.* As the cutoff frequency decreases (center, then left), the rounding becomes more prominent, with the leftmost waveform showing a small ripple.

Question 4.9 $\{251\}$ *What is the average value of each output waveform? The correct answer may surprise you.*

This example also illustrates the impact a lowpass filter can have on a waveform. The simple RC filter used here has a rather gradual frequency response, which means that higher harmonics are smoothly suppressed. Later, we will describe filters that have much more rapidly varying frequency responses, allowing a much more dramatic selection of the input's Fourier coefficients.

More importantly, we have calculated the output of a circuit to a periodic input *without* writing, much less solving, the differential equation governing the circuit's behavior. Furthermore, we made these calculations entirely in the frequency domain. Using Fourier series, we can calculate how *any* linear circuit will respond to a periodic input.

4.5 The Fourier Transform

Fourier series clearly opens the frequency domain as an interesting and useful way of determining how circuits and systems respond to *periodic* input signals. Can we use similar techniques for nonperiodic signals? What is the response of the filter to a single pulse? Addressing these issues requires us to find the Fourier spectrum of all signals, both periodic and nonperiodic ones. We need a definition for *the* Fourier spectrum of a signal, periodic or not. This spectrum is calculated by what is known as the *Fourier transform.*

Let $s_T(t)$ be a periodic signal having period T. We want to consider what happens to this signal's spectrum as we let the period become longer and longer. We denote the spectrum for any assumed value of the period by $c_k^{(T)}$. We calculate the spectrum according to the familiar formula

$$c_k^{(T)} = \frac{1}{T}\int_{-T/2}^{T/2} s_T(t)e^{-j\frac{2\pi kt}{T}}\,dt\ ,$$

where we have used a symmetric placement of the integration interval about the origin for subsequent derivational convenience. Let f be a *fixed* frequency equaling k/T; we vary the frequency index k proportionally as we increase the period.

---

*Later, we shall show that the analytic expression for this shape is the exponential.

Define

$$S_T(f) \equiv T c_k^{(T)} = \int_{-T/2}^{T/2} s_T(t) e^{-j2\pi f t}\, dt\ ,$$

making the corresponding Fourier series

$$s_T(t) = \sum_{k=-\infty}^{\infty} S_T(f) e^{+j2\pi f t} \cdot \frac{1}{T}\ .$$

As the period increases, the spectral lines become closer together, becoming a continuum. Therefore,

$$\lim_{T\to\infty} s_T(t) \equiv \boxed{s(t) = \int_{-\infty}^{\infty} S(f) e^{+j2\pi f t}\, df} \tag{4.13a}$$

$$\text{with} \quad \boxed{S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi f t}\, dt} \tag{4.13b}$$

$S(f)$ is the Fourier transform of $s(t)$—the Fourier transform is symbolically denoted by the uppercase version of the signal's symbol —and is defined for *any* signal for which the integral (4.13b) converges.

Example

Let's calculate the Fourier transform of the pulse signal $\mathrm{p}(t)$ (shown on page 14).

$$\begin{aligned} P(f) &= \int_{-\infty}^{\infty} \mathrm{p}(t) e^{-j2\pi f t} dt \\ &= \int_{0}^{\Delta} e^{-j2\pi f t} dt \\ &= \frac{1}{-j2\pi f}\left[e^{-j2\pi f\Delta} - 1\right] \\ P(f) &= e^{-j\pi f\Delta}\frac{\sin \pi f\Delta}{\pi f} \end{aligned}$$

Note how closely this result resembles the expression for Fourier series coefficients of the periodic pulse signal given by Eq. (4.10) $\{96\}$. Figure 4.10 shows how increasing the period does indeed lead to a continuum of coefficients, and that the Fourier transform does correspond to what the continuum becomes. The quantity $\frac{\sin t}{t}$ has a special name, the *sinc* (pronounced "sink") function, and is denoted by

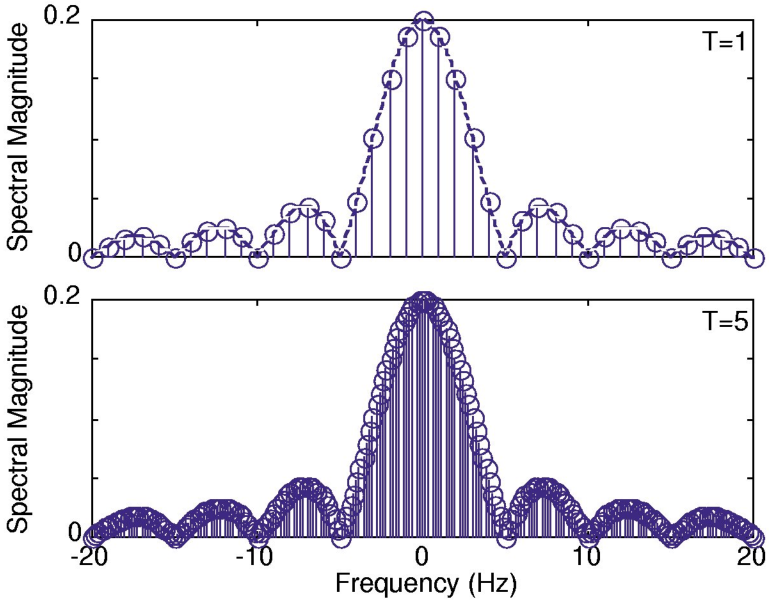


Figure 4.10 The upper plot shows the magnitude of the Fourier series spectrum for the case of $T = 1$ with the Fourier transform of $\mathrm{p}(t)$ show as a dashed line. For the bottom panel, we expanded the period to $T = 5$, keeping the pulse's duration fixed at 0.2, and computed its Fourier series coefficients.

$\operatorname{sinc} t$. Thus, the magnitude of the pulse's Fourier transform equals $|\Delta \operatorname{sinc} \pi f \Delta|$.

The Fourier transform relates a signal's time and frequency domain representations to each other. The direct Fourier transform (or simply the Fourier transform) calculates a signal's frequency domain representation from its time-domain variant (Eq. (4.13b)). The inverse Fourier transform (Eq. (4.13a)) finds the time-domain representation from the frequency domain. Rather than explicitly writing the required integral, we often symbolically express these transform calculations as $\mathcal{F}[s]$ and $\mathcal{F}^{-1}[S]$, respectively.

$$\mathcal{F}[s] = S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft}\, dt$$

$$\mathcal{F}^{-1}[S] = s(t) = \int_{-\infty}^{\infty} S(f) e^{+j2\pi ft}\, df$$

We must have $s(t) = \mathcal{F}^{-1}[\mathcal{F}[s(t)]]$ and $S(f) = \mathcal{F}\left[\mathcal{F}^{-1}[S(f)]\right]$, and these results are indeed valid with minor exceptions.[*] Showing that you "get back to where you started" is difficult from an analytic viewpoint, and we won't try here. Note that the direct and inverse transforms differ only in the sign of the exponent.

Question 4.10 {251} *The differing exponent signs means that some curious results occur when we use the wrong sign. What is* $\mathcal{F}\left[S(f)\right]$*? In other words, use the wrong exponent sign in evaluating the inverse Fourier transform.*

Properties of the Fourier transform and some useful transform pairs are provided in Table 4.1. Especially important among these is *Parseval's Theorem*, which states that power computed in either domain equals the power in the other.

$$\boxed{\int_{-\infty}^{\infty} s^2(t)\,dt = \int_{-\infty}^{\infty} |S(f)|^2\,df} \tag{4.14}$$

Of practical importance is the conjugate symmetry property: When $s(t)$ is real-valued, the spectrum at negative frequencies equals the complex conjugate of the spectrum at the corresponding positive frequencies. Consequently, we need only plot the positive frequency portion of the spectrum (we can easily determine the remainder of the spectrum).

Question 4.11 {251} *How many Fourier transform operations need to be applied to get the original signal back:* $\mathcal{F}\left[\cdots\mathcal{F}\left[s\right]\right] = s$*?*

Note that these mathematical relationships are termed *transforms*. We are transforming (in the nontechnical meaning of the word) a signal from one representation to another. We express Fourier transform *pairs* as $s(t) \leftrightarrow S(f)$. A signal's time and frequency domain representations are uniquely related to each other. A signal thus "exists" in both the time and frequency domains, with the Fourier transform bridging between the two. We can define an information carrying signal in either the time or frequency domains; it behooves the wise engineer to use the simpler of the two.

A common misunderstanding is that while a signal exists in both the time and frequency domains, a single formula expressing a signal must contain *only* time or frequency: Both cannot be present simultaneously. This situation mirrors what happens with complex amplitudes in circuits: As we reveal how communications systems work and are designed, we will define signals entirely in the frequency

---

[*] Recall that the Fourier series for a square wave gives a value for the signal at the discontinuities equal to the average value of the jump. This value may differ from how the signal is *defined* in the time domain, but being unequal at a point is indeed minor.

Short Table of Fourier Transform Pairs

$s(t)$	$S(f)$
$e^{-at}u(t)$	$\dfrac{1}{j2\pi f + a}$
$e^{-a\|t\|}$	$\dfrac{2a}{4\pi^2 f^2 + a^2}$
$\mathrm{p}(t) = \begin{cases} 1, & \|t\| < \Delta/2 \\ 0, & \|t\| > \Delta/2 \end{cases}$	$\dfrac{\sin \pi f \Delta}{\pi f}$
$\dfrac{\sin 2\pi W t}{\pi t}$	$S(f) = \begin{cases} 1, & \|f\| < W \\ 0, & \|f\| > W \end{cases}$

Fourier Transform Properties

	Time-Domain	Frequency Domain
Linearity	$a_1 s_1(t) + a_2 s_2(t)$	$a_1 S_1(f) + a_2 S_2(f)$
Conjugate Symmetry	$s(t)$ real	$S(f) = S^*(-f)$
Even Symmetry	$s(t) = s(-t)$	$S(f) = S(-f)$
Odd Symmetry	$s(t) = -s(-t)$	$S(f) = -S(-f)$
Scale Change	$s(at)$	$\dfrac{1}{\|a\|} S(f/a)$
Time Delay	$s(t - \tau)$	$e^{-j2\pi f \tau} S(f)$
Complex Modulation	$e^{j2\pi f_0 t} s(t)$	$S(f - f_0)$
Amplitude Modulation	$s(t) \cos 2\pi f_0 t$	$\dfrac{S(f - f_0) + S(f + f_0)}{2}$
	$s(t) \sin 2\pi f_0 t$	$\dfrac{S(f - f_0) - S(f + f_0)}{2j}$
Differentiation	$\dfrac{d\,s(t)}{dt}$	$j2\pi f S(f)$
Integration	$\displaystyle\int_{-\infty}^{t} s(\alpha)\,d\alpha$	$\dfrac{1}{j2\pi f} S(f)$ if $S(0) = 0$
Multiplication by t	$t s(t)$	$\dfrac{1}{-j2\pi} \dfrac{d\,S(f)}{df}$
Area	$\displaystyle\int_{-\infty}^{\infty} s(t)\,dt$	$S(0)$
Value at origin	$s(0)$	$\displaystyle\int_{-\infty}^{\infty} S(f)\,df$
Parseval's Theorem	$\displaystyle\int_{-\infty}^{\infty} \|s(t)\|^2\,dt$	$\displaystyle\int_{-\infty}^{\infty} \|S(f)\|^2\,df$

Table 4.1 Fourier transform properties and relations.

domain without explicitly finding their time domain variants. We did that in the previous section (§4.4) when we defined Fourier series coefficients according to letter to be transmitted. Thus, a signal, though most familiarly defined in the time-domain, really can be defined equally as well (and sometimes more easily) in the frequency domain. They depend on frequency, and the time variable cannot appear in impedances.

As this chapter continues, we will learn that finding a linear, time-invariant system's output in the time domain can be most easily calculated by determining the input signal's spectrum, performing a simple calculation in the frequency domain, and inverse transforming the result. Furthermore, understanding communications and information processing systems requires a thorough understanding of signal structure and of how systems work in *both* the time and frequency domains.

The only difficulty in calculating the Fourier transform of any signal occurs when we have periodic signals (in either domain). Realizing that the Fourier series is a special case of the Fourier transform, we simply calculate the Fourier series coefficients instead, and plot them along with the spectra of nonperiodic signals on the same frequency axis.

Example

In communications, a very important operation on a signal $s(t)$ is to *amplitude modulate* it. Using this operation more as an example rather than elaborating the communications aspects here, we want to compute the Fourier transform —the spectrum—of $[1+s(t)]\cdot\cos 2\pi f_c t$. Thus,

$$x(t)=[1+s(t)]\cdot\cos 2\pi f_c t=\cos 2\pi f_c t+s(t)\cos 2\pi f_c t\ .$$

For the spectrum of $\cos 2\pi f_c t$, we use the Fourier series. Its period is $1/f_c$, and its only nonzero Fourier coefficients are $c_{\pm 1}=\frac{1}{2}$.* The second term is *not* periodic unless $s(t)$ has the same period as the sinusoid. Using Euler's relation, the spectrum of the second term can be derived as

$$\begin{aligned}
s(t)\cos 2\pi f_c t &= \left(\int_{-\infty}^{\infty} S(f)e^{j2\pi ft}\,df\right)\cos 2\pi f_c t \\
&= \frac{1}{2}\int_{-\infty}^{\infty} S(f)e^{j2\pi(f+f_c)t}\,df+\frac{1}{2}\int_{-\infty}^{\infty} S(f)e^{j2\pi(f-f_c)t}\,df \\
&= \frac{1}{2}\int_{-\infty}^{\infty} S(f-f_c)e^{j2\pi ft}\,df+\frac{1}{2}\int_{-\infty}^{\infty} S(f+f_c)e^{j2\pi ft}\,df \\
&= \int_{-\infty}^{\infty}\frac{S(f-f_c)+S(f+f_c)}{2}e^{j2\pi ft}\,df
\end{aligned}$$

---

*From Euler's relation, $\cos 2\pi f_c t=\frac{1}{2}e^{-j2\pi f_c t}+\frac{1}{2}e^{+j2\pi f_c t}$. This is the Fourier series for cosine!

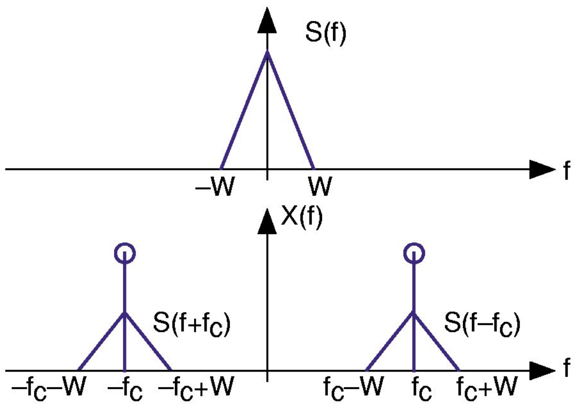


Figure 4.11 A signal have a triangular shaped spectrum is shown in the top plot. Its highest frequency—the largest frequency containing power—is W Hz. Once amplitude modulated, the resulting spectrum has "lines" corresponding to the Fourier series components at $\pm f_c$ and the original triangular spectrum shifted to $\pm f_c$ and scaled by $1/2$.

Exploiting the uniqueness property of the Fourier transform, we have

$$\mathcal{F}\left[s(t)\cos 2\pi f_c t\right] = \frac{S(f-f_c)+S(f+f_c)}{2} \ .$$

This component of the spectrum consists of the original signal's spectrum delayed and advanced *in frequency*. The spectrum of the amplitude modulated signal is shown in Fig. 4.11. Note how in this figure the signal $s(t)$ is defined in the frequency domain. To find its time domain representation, we simply use the inverse Fourier transform.

Question 4.12 {**252**} *What is the signal $s(t)$ that corresponds to the spectrum shown in the upper panel of Figure 4.11 {108}?*

Question 4.13 {**252**} *What is the power in $x(t)$, the amplitude-modulated signal? Try the calculation in both the time and frequency domains.*

In this example, we call the signal $s(t)$ a *baseband signal* because its power is contained at low frequencies. Signals such as speech and the Dow Jones averages are baseband signals. The baseband signal's *bandwidth* equals W Hz, the highest frequency at which it has power. Since $x(t)$'s spectrum is confined to a frequency band not close to the origin (we assume $f_c \gg W$), we have a *bandpass signal*. The bandwidth of a bandpass signal is *not* its highest frequency, but the range of positive frequencies where the signal has power. Thus, in this example, the bandwidth is $2W$ Hz. Why a signal's bandwidth should depend on its spectral shape will become clear once we develop communications systems.

4.6 Linear, Time-Invariant Systems

We showed that when we apply a periodic input to a linear, time-invariant system, the output is periodic and has Fourier series coefficients equal to the product of the system's frequency response and the input's Fourier coefficients (Eq. (4.12) $\{100\}$). The way we derived the spectrum of non-periodic signal from periodic ones makes it clear that the same kind of result works when the input is not periodic: *If $x(t)$ serves as the input to a linear, time-invariant system having frequency response $H(f)$, the spectrum of the output is $X(f)H(f)$.*

Example

Let's use this frequency-domain input-output relationship for linear, time-invariant systems to find a formula for the RC-circuit's response to a pulse input. We have expressions for the input's spectrum and the system's frequency response.

$$\begin{aligned} P(f) &= e^{-j\pi f\Delta}\frac{\sin \pi f\Delta}{\pi f} \\ H(f) &= \frac{1}{1+j2\pi fRC} \end{aligned}$$

Thus, the output's Fourier transform equals

$$Y(f) = e^{-j\pi f\Delta}\frac{\sin \pi f\Delta}{\pi f}\cdot\frac{1}{1+j2\pi fRC}$$

You won't find this Fourier transform in our table, and the required integral is difficult to evaluate as the expression stands. This situation requires cleverness and an understanding of the Fourier transform's properties. In particular, recall Euler's relation for the sinusoidal term and note the fact that multiplication by a complex exponential in the frequency domain amounts to a time delay. Let's momentarily make the expression for $Y(f)$ more complicated.

$$\begin{aligned} e^{-j\pi f\Delta}\frac{\sin \pi f\Delta}{\pi f} &= e^{-j\pi f\Delta}\frac{e^{+j\pi f\Delta}-e^{-j\pi f\Delta}}{j2\pi f} \\ &= \frac{1}{j2\pi f}\cdot\left[1-e^{-j2\pi f\Delta}\right] \end{aligned} \tag{4.15}$$

Consequently,

$$Y(f) = \frac{1}{j2\pi f} \cdot \left[1 - e^{-j2\pi f\Delta}\right] \cdot \frac{1}{1 + j2\pi fRC} \tag{4.16}$$

The table of Fourier transform properties (Table 4.1 $\{106\}$) suggests thinking about this expression as a *product* of terms.

- Multiplication by $1/j2\pi f$ means integration.
- Multiplication by the complex exponential $e^{-j2\pi f\Delta}$ means delay by Δ seconds in the time domain.
- The term $\left[1 - e^{-j2\pi f\Delta}\right]$ means, in the time domain, subtract from the original signal its time-delayed version.
- The inverse transform of the frequency response is $\frac{1}{RC}e^{-t/RC}\,\mathrm{u}(t)$.

We can translate each of these frequency-domain products into time-domain operations *in any order we like* because the order in which multiplications occur doesn't affect the result. Let's start with the product of $1/j2\pi f$ (integration in the time domain) and the transfer function.

$$\frac{1}{j2\pi f} \cdot \frac{1}{1 + j2\pi fRC} \longleftrightarrow \left(1 - e^{-t/RC}\right)\mathrm{u}(t)\ .$$

The middle term in the expression for $Y(f)$ consists of the difference of two terms: the constant 1 and the complex exponential $e^{-j2\pi f\Delta}$. Because of the Fourier transform's linearity, we simply subtract the results.

$$Y(f) \longleftrightarrow \left(1 - e^{-t/RC}\right)\mathrm{u}(t) - \left(1 - e^{-(t-\Delta)/RC}\right)\mathrm{u}(t - \Delta)$$

Note that in delaying the signal how we carefully included the unit step. The second term in this result does not begin until $t = \Delta$. Thus, the waveforms of Figure 4.9 $\{101\}$ are exponentials. We say that the *time constant* $\{13\}$ of an exponential decaying signal equals the time it takes to decrease by $1/e$ of its original value. Thus, the time-constant of the rising and falling portions of the output equal the product of the circuit's resistance and capacitance.

Question 4.14 $\{\mathbf{252}\}$ *Derive the filter's output by considering the terms in* (4.15) $\{$*109*$\}$ *in the order given. Integrate last rather than first. You should get the same answer.*

In this example, we used the table extensively to find the inverse Fourier transform, relying mostly on what multiplication by certain factors, like $1/j2\pi f$ and $e^{-j2\pi f\Delta}$, meant. We essentially treated multiplication by these factors as if they were transfer functions of some fictitious circuit. The transfer function $1/j2\pi f$ corresponded to a circuit that integrated, and $e^{-j2\pi f\Delta}$ to one that delayed. We even implicitly interpreted the circuit's transfer function as the input's spectrum! This approach to finding inverse transforms —breaking down a complicated expression into products and sums of simple components —is the engineer's way of breaking down the problem into several much simpler ones. Along the way we may make the system serve as the input, but in the rule $Y(f) = X(f)H(f)$, which term is the input and which is the transfer function is merely a notational matter (we labeled one factor with an X and the other with an H).

The notion of a transfer function applies well beyond linear circuits. Although we don't have all we need to demonstrate the result as yet, *all* linear, time-invariant systems have a frequency-domain input-output relation given by the product of the input's Fourier transform and the system's transfer function. Thus, linear circuits are a special case of linear, time-invariant systems. As we tackle more sophisticated problems in transmitting, manipulating, and receiving information, we will assume linear systems having certain properties (transfer functions) *without* worrying about what circuit has the desired property. At this point, you may be concerned that this approach is glib, and rightly so. Later we'll show that by involving software that we really don't need to be concerned in most cases.

Another interesting notion arises from the commutative property of multiplication we exploited in the example: We rather arbitrarily chose an order in which to apply each product. Consider a cascade of two linear, time-invariant systems. Because the Fourier transform of the first system's output is $X(f)H_1(f)$ and it serves as the second system's input, the cascade's output spectrum is $X(f)H_1(f)H_2(f)$. Because this product also equals $X(f)H_2(f)H_1(f)$, the *cascade having the linear systems in the opposite order yields the same result.* Furthermore, the cascade acts like a *single* linear system, having transfer function $H_1(f)H_2(f)$. This result applies to other configurations of linear, time-invariant systems as well; see Problem 4.10 $\{128\}$. Engineers exploit this property by determining what transfer function they want, then breaking it down into components arranged according to standard configurations. Using the fact that op-amp circuits can be connected in cascade with the transfer function equaling the product of its component's transfer function (Problem 3.19 $\{80\}$), we find a ready way of realizing designs. We now understand why op-amp implementations of transfer functions are so important.

4.7 Modeling the Speech Signal

The information contained in the spoken word is conveyed by the speech signal. Because we shall analyze several speech transmission and processing schemes, we need to understand the speech signal's structure —what's special about the speech signal—and how we can describe, *model*, speech production. This modeling effort consists of finding a system's description of how relatively unstructured signals, arising from simple sources, are given structure by passing them through an interconnection of systems to yield speech. For speech and for many other situations, system choice is governed by the physics underlying actual production process. Because the fundamental equation of acoustics—the wave equation—applies here and is linear, we can use linear systems in our model with a fair amount of accuracy.*

Fig. 4.12 shows the actual and model speech production systems. The model differs whether you are saying a vowel or a consonant. We concentrate first on the vowel production mechanism. When the vocal cords are placed under tension by the surrounding musculature, air pressure from the lungs causes the vocal cords to vibrate. To visualize this effect, take a rubber band and hold it in front of your lips. If held open when you blow through it, the air passes through more or less freely; this situation corresponds to "breathing mode." If held tautly and close together, blowing through the opening causes the sides of the rubber band to vibrate. † You can imagine what the airflow is like on the opposite side of the rubber band or the vocal cords. Your lung power is the simple source referred to earlier; it can be modeled as a constant supply of air pressure. The vocal cords respond to this input by vibrating, which means the output of this system is some periodic function.

Question 4.15 {252} *Note that the vocal cord system takes a constant input and produces a periodic airflow that corresponds to its output signal. Is this system linear or nonlinear? Justify your answer.*

The period of the vocal cord's output for vowels is known as the *pitch*. Indeed, singers modify vocal cord tension to change the pitch to produce the desired musical note. Vocal cord tension is governed by a control input to the musculature; in

---

*The naturalness of linear system models for speech does not extend to other situations. In many cases, the underlying mathematics governing by the physics, biology, and/or chemistry of the problem are nonlinear, leaving linear systems models as approximations. Nonlinear models are far more difficult at the current state of knowledge to understand, and information engineers frequently prefer linear models because they provide a greater level of comfort, but not necessarily a sufficient level of accuracy.

†This effect works best with a wide rubber band.

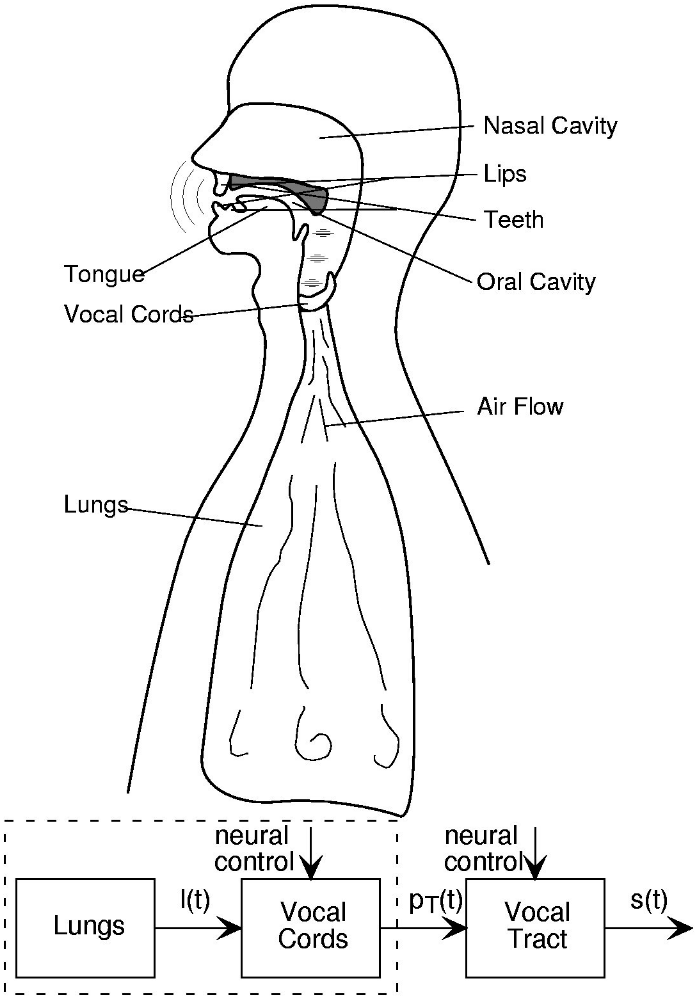

Nasal Cavity
Lips
Teeth
Oral Cavity
Tongue
Vocal Cords
Air Flow
Lungs
neural control
neural control
Lungs
l(t)
Vocal Cords
pT(t)
Vocal Tract
s(t)

Figure 4.12 The vocal tract is shown in cross-section on the top, and the systems model for it below. Air pressure produced by the lungs force air through the vocal cords that, when under tension, produce puffs of air that excite resonances in the vocal and nasal cavities. What are not shown are the brain and the musculature that control the entire speech production process. The signals $l(t)$, $p_T(t)$, and $s(t)$ are the air pressure provided by the lungs, the periodic pulse output provided by the vocal cords, and the speech output respectively. Control signals from the brain are shown as entering the systems from the top. Clearly, these come from the same source, but for modeling purposes we describe them separately since they control different aspects of the speech signal.

system's models we represent control inputs as signals coming into the top or bottom of the system. Certainly in the case of speech and in many other cases as well, it is the control input that carries information, impressing it on the system's output. The resulting change of signal structure resulting from varying the control input enables information to be conveyed by the signal, a process generically known as *modulation*. In singing, musicality is largely conveyed by pitch; in western speech, it is much less important.[‡] A sentence can be read in a monotone fashion without completely destroying the information expressed by the sentence. However, the difference between a statement and a question is frequently expressed by pitch changes. For example, note the sound differences between "Let's go to the park." and "Let's go to the park?"

For some consonants, the vocal cords vibrate just as in vowels. For example, the so-called nasal sounds "n" and "m" have this property. For others, the vocal cords do not produce a periodic output. Going back to mechanism, when consonants such as "f" are produced, the vocal cords are placed under much less tension, which results in turbulent flow. (You can emulate this situation with your rubber band.) The resulting output airflow is quite erratic, so much so that we describe it as being *noise*. We define noise carefully later when we delve into communication problems.

The vocal cords' periodic output can be well described by the periodic pulse train $\mathrm{p}_T(t)$, with T denoting the pitch period. The spectrum of this signal contains harmonics of the frequency $1/T$, what is known as the *pitch frequency* or the *fundamental frequency* $F0$. The primary difference between adult male and female/prepubescent speech is pitch. Before puberty, pitch frequency for normal speech ranges between 150–400 Hz for both males and females. After puberty, the vocal cords of males undergo a physical change, which has the effect of lowering their pitch frequency to the range 80–160 Hz. If we could examine the vocal cord output, we could probably discern whether the speaker was male or female. As we will soon learn, this difference is also readily apparent in the speech signal itself.

To simplify our speech modeling effort, we shall assume that the pitch period is constant. With this simplification, we collapse the vocal-cord–lung system as a simple source that produces the periodic pulse signal (Figure 4.12 $\{113\}$). The sound pressure signal thus produced enters the mouth behind the tongue, creates acoustic disturbances, and exits primarily through the lips and to some extent through the nose. Speech specialists tend to name the mouth, tongue, teeth, lips, and nasal cavity the *vocal tract*. The physics governing the sound disturbances produced in the vocal tract and those of an organ pipe are quite similar. Whereas the organ pipe has the simple physical structure of a straight tube, the cross-section of the vocal

---

[‡] In oriental languages, pitch plays a much larger role.

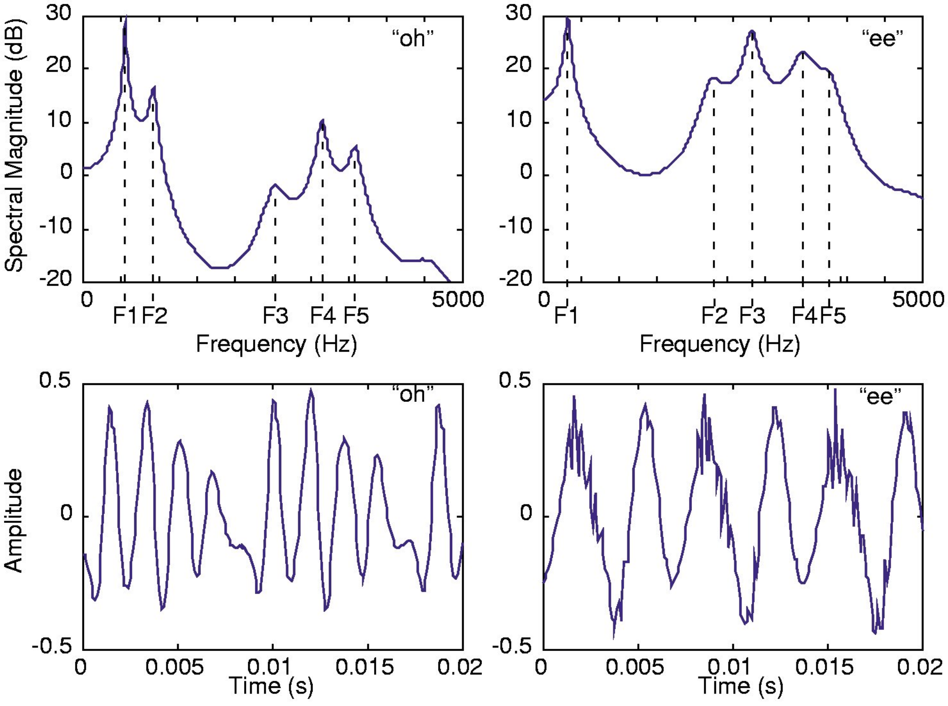


Figure 4.13 The ideal frequency response of the vocal tract as it produces the sounds "oh" and "ee" are shown on the top left and top right, respectively. The spectral peaks are known as formants, and are numbered consecutively from low to high frequency. The bottom plots show speech waveforms corresponding to these sounds.

tract varies along its length because of the positions of the tongue, teeth, and lips. It is these positions that are controlled by the brain to produce the vowel sounds. Spreading the lips, bringing the teeth together, and bringing the tongue toward the front portion of the roof of the mouth produces the sound ee. Rounding the lips, spreading the teeth, and positioning the tongue toward the back of the oral cavity produces the sound oh. These variations result in a linear, time-invariant system that has a frequency response typified by several peaks, as shown in Fig. 4.13. These peaks occur are known as *formants*. Thus, speech signal processors would say that the sound "oh" has a higher first formant frequency than the sound "ee", with $F2$ being much higher during "ee." $F2$ and $F3$ (the second and first formants) have more energy in "ee" than in "oh." Rather than serving as a filter, rejecting high or low frequencies, the vocal tract serves to *shape* the spectrum of the vocal cords.

In the time domain, we have a periodic signal, the pitch, serving as the input to a linear system. We know that the output —the speech signal we utter and that is heard by others and ourselves —will also be periodic. Example time-domain speech signals are shown in Fig. 4.13, where the periodicity is quite apparent.

Question 4.16 {**252**} *From the waveform plots shown in Fig. 4.13, determine the pitch period and the pitch frequency.*

Consequently, speech has a Fourier series representation given by Eq. (4.12) {100}. Because the acoustics of the vocal tract are linear, we know that the spectrum of the output equals the product of the pitch signal's spectrum and the vocal tract's frequency response. We thus obtain the *fundamental model of speech production.*

$$\boxed{S(f) = P_T(f) \cdot H_V(f)} \tag{4.17}$$

Here, $H_V(f)$ is the transfer function of the vocal tract system. The Fourier series for the vocal cords' output is given by Eq. (4.10) {96} and is plotted in Figure 4.7 {97}. If we had, for example, a male speaker with about a 150 Hz pitch ($T \approx 6.7$ ms) saying the vowel "oh", the spectrum of his speech *predicted by our model* would be as shown in Fig. 4.14. The model spectrum idealizes the measured spectrum, and captures all the important features. The measured spectrum certainly demonstrates what are known as *pitch lines*, and we realize from our model that they are due to the vocal cord's periodic excitation of the vocal tract. The vocal tract's shaping of the line spectrum is clearly evident, but difficult to discern exactly. The model transfer function for the vocal tract makes the formants much more readily evident.

Question 4.17 {**252**} *The Fourier series coefficients for speech are related to the vocal tract's transfer function* only *at the frequencies* k/T, $k = 0, 1, 2, \ldots$; *see Eq.* (4.10) {*96*}. *Would male or female speech tend to have a more clearly identifiable formant structure when its spectrum is computed? Consider, for example, how the spectrum shown in Fig. 4.14 would change if the pitch were twice as high (*$\sim$ 300 *Hz).*

When we speak, pitch and the vocal tract's transfer function are not static; they change according to their control signals to produce speech. Engineers typically display how the speech spectrum changes over time with what is known as a *spectrogram* (Fig. 4.15). Note how the line spectrum, which indicates how the pitch changes, is visible during the vowels, but not during the consonants (like the *ce* in "Rice").

The fundamental model for speech indicates how engineers use the physics underlying the signal generation process and exploit its structure to produce a

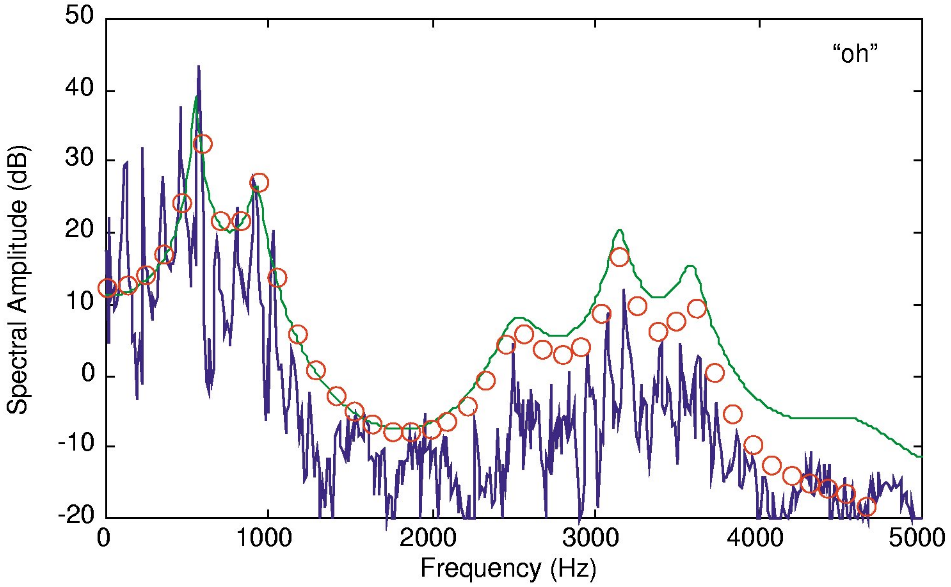


Figure 4.14 The vocal tract's transfer function, shown as the thin, smooth line, is superimposed on the spectrum of actual male speech corresponding to the sound "oh." The spectrum of the model's output is shown as the sequence of circles, which indicate the amplitude and frequency of each Fourier series coefficient.

systems model that suppresses the physics while emphasizing how the signal is "constructed." From everyday life, we know that speech contains a wealth of information. We want to determine how to transmit and receive it. Efficient and effective speech transmission requires us to know the signal's properties and its structure (as expressed by the fundamental model of speech production). We see from Fig. 4.15, for example, that speech contains significant energy from zero frequency up to around 5 kHz. *Effective* speech transmission systems must be able to cope with signals having this bandwidth. It is interesting that one system that does not support this 5 kHz bandwidth is the telephone: Telephone systems act like a *bandpass filter* passing energy between about 200 Hz and 3.2 kHz. The most important consequence of this filtering is the removal of high frequency energy. In our sample utterance, the "ce" sound in "Rice" contains most of its energy above 3.2 kHz; this filtering effect is why it is extremely difficult to distinguish the sounds "s" and "f" over the telephone. Radio does support this bandwidth; we learn more about AM and FM radio systems later. *Efficient* speech transmission systems exploit the speech signal's special structure: What makes speech speech? You can conjure many signals that span the same frequencies as speech —car en-

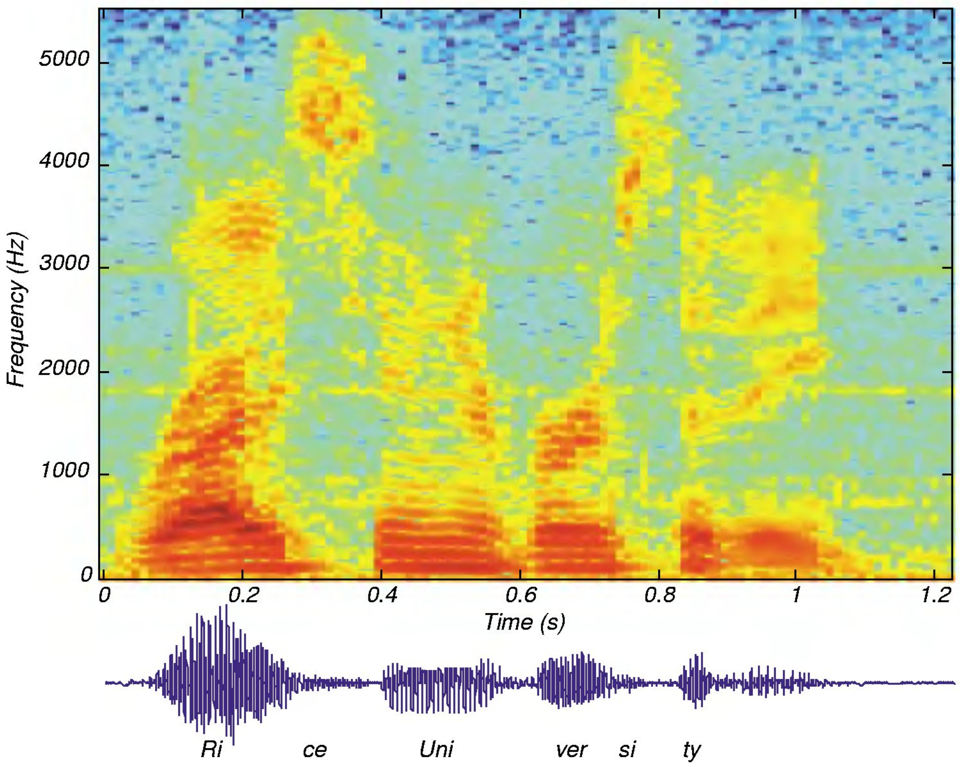


Figure 4.15 Displayed is the spectrogram of one of the authors saying "Rice University." Blue indicates low energy portion of the spectrum, with red indicating the most energetic portions. Below the spectrogram is the time-domain speech signal, where the periodicities are clearly evident.

---

gine sounds, violin music, dog barks —but don't sound at all like speech. We shall learn later that transmission of *any* 5 kHz bandwidth signal requires about 80 kbps (thousands of bits per second) to transmit digitally. *Speech* signals can be transmit using less than 1 kbps because of its special structure. Reducing the "digital bandwidth" so drastically took engineers many years to develop signal processing and coding methods that could deal effectively with speech without destroying it. If you used a speech transmission system to send a violin sound, it would arrive horribly distorted; speech transmitted the same way would sound fine.

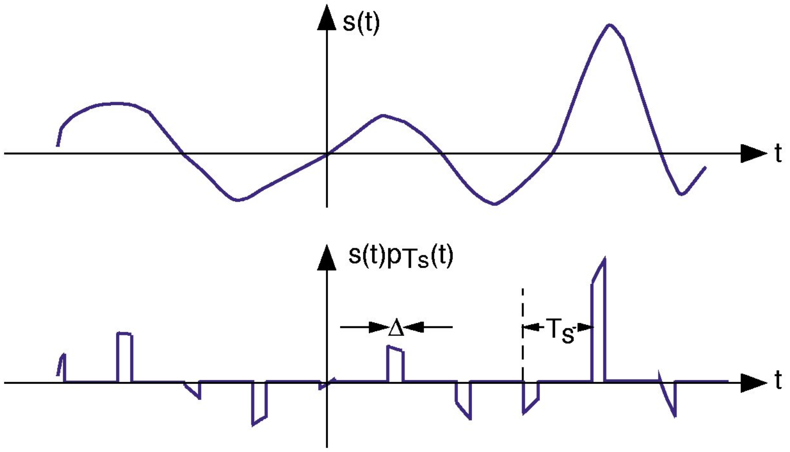


Figure 4.16 The waveform of an example signal is shown in the top plot and its sampled version in the bottom.

4.8 The Sampling Theorem

Digital transmission of information and digital signal processing all require signals to first be "acquired" by a computer. One of the most amazing and useful results in electrical engineering is that signals can be converted from a function of time into a sequence of numbers *without error*: We can convert the numbers back into the signal with (theoretically) *no* error. Harold Nyquist, a Bell Laboratories engineer, derived this result, known as the Sampling Theorem, first in the 1920s. It found no real application back then. Claude Shannon, also at Bell Laboratories, revived the result once computers were made public after World War II.

We begin with the periodic pulse signal $\mathrm{p}_{T_s}(t)$, this time centered about the origin so that it's Fourier series coefficients are real (the signal is even).

$$\mathrm{p}_{T_s}(t) = \sum_{k=-\infty}^{\infty} c_k e^{\frac{j2\pi kt}{T_s}}, \quad c_k = \frac{\sin \frac{\pi k\Delta}{T_s}}{\pi k} \tag{4.18}$$

To *sample* the signal $s(t)$, we multiply it by the periodic pulse signal. The resulting signal $x(t) = s(t)\,\mathrm{p}_{T_s}(t)$, as shown in Fig. 4.16, has nonzero values only during the time intervals $(nT_s - \Delta/2, nT_s + \Delta/2)$, $n = \ldots, -1, 0, 1, \ldots$. If the properties of $s(t)$ and the periodic pulse signal are chosen properly, we can recover $s(t)$ from $x(t)$ by filtering.

To understand how signal values between the samples can be "filled" in, we need to calculate the sampled signal's spectrum. Using the Fourier series represen-

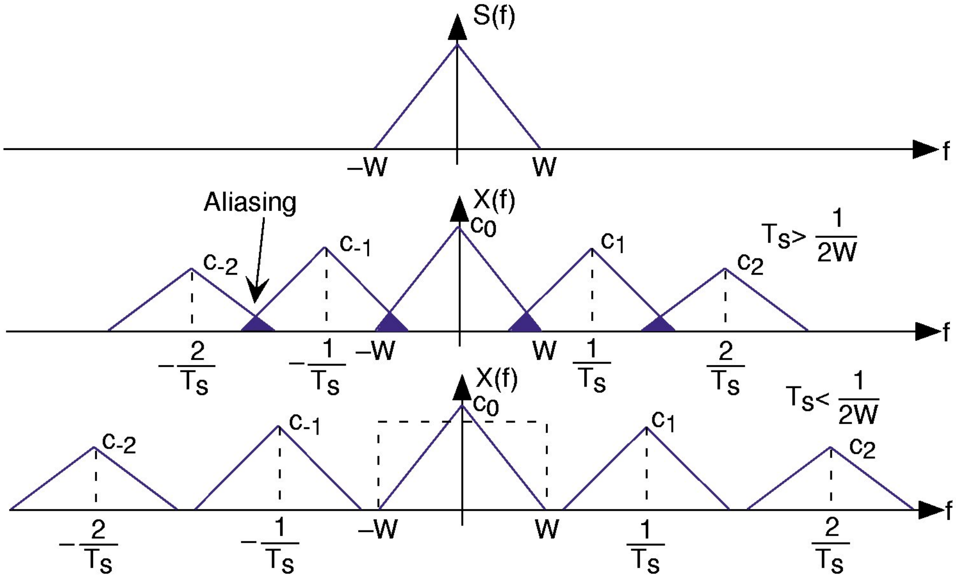


Figure 4.17 The spectrum of some bandlimited (to W Hz) signal is shown in the top plot. If the sampling interval T_s is chosen too large relative to the bandwidth W, aliasing will occur. In the bottom plot, the sampling interval is chosen sufficiently small to avoid aliasing. Note that if the signal were not bandlimited, the component spectra would *always* overlap.

tation of the periodic sampling signal,

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{\frac{j2\pi kt}{T_s}} s(t) \ .$$

Considering each term in the sum separately, we need to know the spectrum of the product of the complex exponential and the signal. Evaluating this transform directly is quite easy.

$$\begin{aligned}\int_{-\infty}^{\infty} s(t) e^{\frac{j2\pi kt}{T_s}} \cdot e^{-j2\pi ft}\, dt &= \int_{-\infty}^{\infty} s(t) e^{-j2\pi\left(f-\frac{k}{T_s}\right)t}\, dt \\ &= S\left(f-\frac{k}{T_s}\right)\end{aligned}$$

Thus, the spectrum of the sampled signal consists of weighted (by the coefficients c_k) and delayed versions of the signal's spectrum (Fig. 4.17).

$$X(f) = \sum_{k=-\infty}^{\infty} c_k S\left(f - \frac{k}{T_s}\right)$$

In general, the terms in this sum overlap each other in the frequency domain, rendering recovery of the original signal impossible. This unpleasant phenomenon is known as *aliasing*. If, however, we satisfy two conditions:

- The signal $s(t)$ is *bandlimited*—has power in a restricted frequency range—to W Hz, and
- the sampling interval T_s is small enough so that the individual components in the sum do not overlap —$T_s < 1/2W$,

aliasing will not occur. In this delightful case, we can recover the original signal by lowpass filtering $x(t)$ with a filter having a cutoff frequency equal to W Hz. These two conditions ensure the ability to recover a bandlimited signal from its sampled version: We thus have the *Sampling Theorem.*

Question 4.18 {**253**} *The Sampling Theorem (as stated) does not mention the pulse width* Δ. *What is the effect of this parameter on our ability to recover a signal from its samples (assuming the Sampling Theorem's two conditions are met)?*

The frequency $1/2T_s$, known today as the *Nyquist frequency* and the *Shannon sampling frequency*, corresponds to the highest frequency at which a signal can contain energy and remain compatible with the Sampling Theorem. High-quality sampling systems ensure that no aliasing occurs by unceremoniously lowpass filtering the signal (cutoff frequency being slightly lower than the Nyquist frequency) before sampling. Such systems therefore vary the *anti-aliasing* filter's cutoff frequency as the sampling rate varies. Because such quality features cost money, many sound cards do *not* have anti-aliasing filters or, for that matter, post-sampling filters. They sample at high frequencies, 44.1 kHz for example, and hope the signal contains no frequencies above the Nyquist frequency (22.05 kHz in our example). As discovered in Chapter 5 {131}, if no aliasing occurs, we can use digital signal processing to reduce the sampling rate without incurring further aliasing. If, however, the signal contains frequencies beyond the sound card's Nyquist frequency, the resulting aliasing can be impossible to remove.

Question 4.19 {**253**} *To gain a better appreciation of aliasing, sketch the spectrum of a sampled square wave. For simplicity consider only the spectral repetitions centered at* $-1/T_s, 0, 1/T_s$. *Let the sampling interval* T_s *be 1; consider two values for the square wave's period: 3.5 and 4. Note in particular where the spectral lines go as the period decreases; some will move to the left and some to the right. What property characterizes the ones going the same direction?*

If we satisfy the Sampling Theorem's conditions, the signal will change only slightly during each pulse. As we narrow the pulse, making Δ smaller and smaller, the nonzero values of the signal $s(t)\,\mathrm{p}_{T_s}(t)$ will simply be $s(nT_s)$, the signal's *samples*. If indeed the Nyquist frequency equals the signal's highest frequency, at least two samples will occur within the period of the signal's highest frequency sinusoid. In these ways, the sampling signal captures the sampled signal's temporal variations in a way that leaves all the original signal's structure intact.

Question 4.20 {253} *What is the simplest bandlimited signal? Using this signal, convince yourself that less than two samples/period will not suffice to specify it. If the sampling rate* $1/T_s$ *is not high enough, what signal would your resulting undersampled signal become?*

4.9 Analog-to-Digital Conversion

The Sampling Theorem says that if we sample a bandlimited signal $s(t)$ fast enough, it can be recovered without error from its samples $s(nT_s)$, $n = \ldots, -1, 0, 1, \ldots$. Sampling is only the first phase of acquiring data into a computer: Computational processing further requires that the samples be *quantized*: analog values are converted into digital form (§1.2.2 {4}). In short, we will have performed *analog-to-digital* (A/D) *conversion*.

Question 4.21 {253} *Recalling the plot of average daily highs in Problem 4.3 {125}, why is this plot so jagged? Interpret this effect in terms of analog-to-digital conversion.*

A phenomenon reminiscent of the errors incurred in representing numbers on a computer prevents signal amplitudes from be converted with no error into a binary number representation. In analog-to-digital conversion, the signal is assumed to lie within a predefined range. Assuming we can scale the signal without affecting the information it expresses, we'll define this range to be $[-1, 1]$. Furthermore, the A/D converter assigns amplitude values in this range to a set of integers. A B-bit converter produces one of the integers $0, 1, \ldots, 2^B - 1$ for each sampled input. Fig. 4.18 shows how a two-bit A/D converter assigns input values to the integers. We define a *quantization interval* to be the range of values assigned to the same integer. Thus, for our example two-bit A/D converter, the quantization interval is 0.5; in general, it is $2/2^B$. Because values lying anywhere within a quantization interval are assigned the same value for computer processing, *the original amplitude value cannot be recovered without error.* Typically, the D/A converter, the device that converts integers to amplitudes, assigns an amplitude equal to the value lying

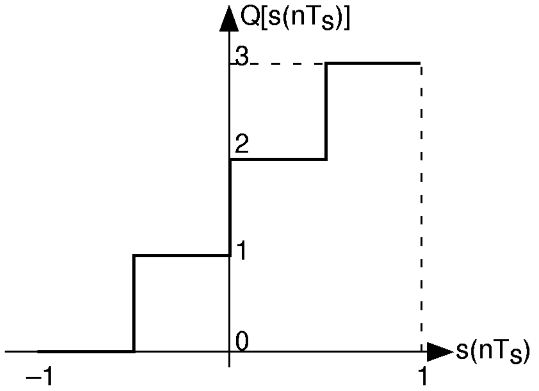


Figure 4.18 A two-bit A/D converter assigns voltage in the range $[-1, 1]$ to one of four possible integers. For example, all inputs having values lying between -1 and -0.5 are assigned the value zero.

halfway in the quantization interval. The integer 0 would be assigned to the amplitude -0.75 in this scheme. The error introduced by converting a signal from analog to digital form by sampling and amplitude quantization then back again would be half the quantization interval for each amplitude value. Thus, the so-called A/D error equals half the width of a quantization interval: $1/2^B$. As we have fixed the input-amplitude range, the more bits available in the A/D converter, the smaller the quantization error.

Question 4.22 {**254**} *How many bits would be required in the A/D converter to ensure that the maximum amplitude quantization error was less than 60 db smaller than the signal's peak value?*

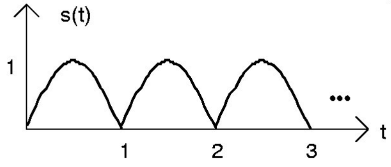
Once we have acquired signals with an A/D converter, we can process them using digital hardware or software. It can be shown that if the computer processing is linear, the result of sampling, computer processing, and unsampling is equivalent to some analog linear system. Why go to all the bother if the same function can be accomplished using analog techniques? The following chapters provide insights into when digital processing excels and when it does not.

Problems

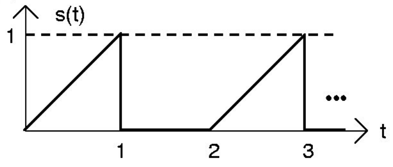
4.1 Fourier Series

Find the Fourier series representation for the following periodic signals.

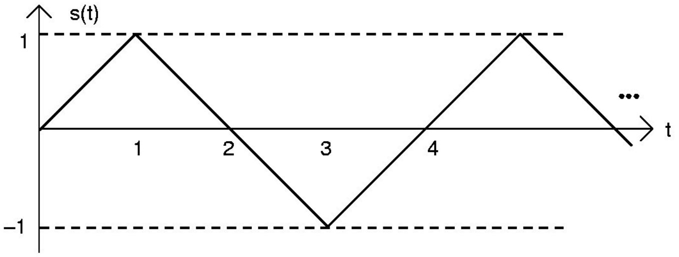
(a)



(b)



(c) Find the complex Fourier series for the triangle wave without performing the usual Fourier integrals. Hint: How is this signal related to one for which you already have the series?

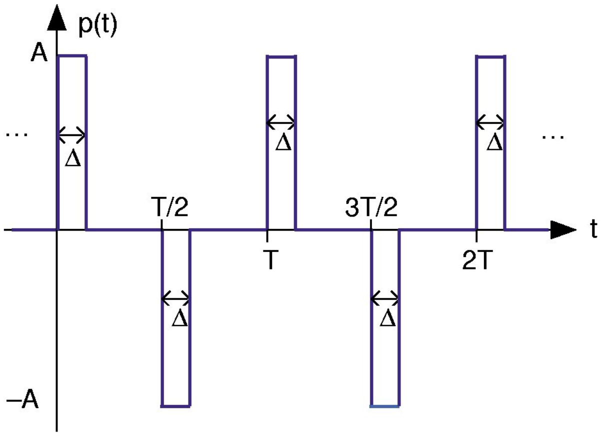


4.2 Fourier Series

Using the properties of the Fourier series can ease finding a signal's spectrum.

(a) Suppose a signal $s(t)$ is periodic with period T. If c_k represents the signal's Fourier series coefficients, what are the Fourier series coefficients of $s(t - T/2)$?

(b) Find the Fourier series of the signal $p(t)$ shown below.

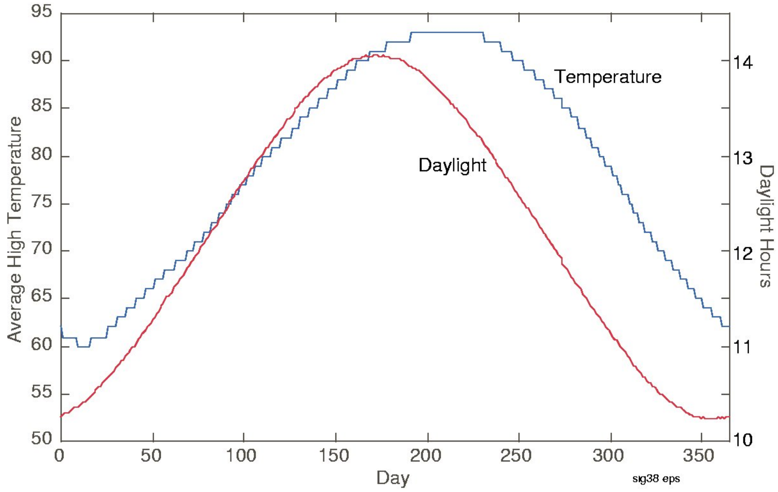


(c) Suppose this signal is used to sample a signal bandlimited to $1/T$ Hz. Find an expression for and sketch the spectrum of the sampled signal.

(d) Does aliasing occur? If so, can a change in sampling rate prevent aliasing; if not, show how the signal can be recovered from these samples.

4.3 Long, Hot Days

The daily temperature is a consequence of several effects, one of them being the sun's heating. If this were the dominant effect, then daily temperatures would be proportional to the number of daylight hours. The plot shows that the average daily high temperature does *not* behave that way.



In this problem, we want to understand the temperature component of our environment using Fourier series and linear system theory. The file `temperature.mat` contains these data (daylight hours in the first row, corresponding average daily highs in the second) for Houston, Texas.

(a) Let the length of day serve as the sole input to a system having an output equal to the average daily temperature. Examining the plots of input and output, would you say that the system is linear or not? How did you reach your conclusion?

(b) Find the first five terms ($c_0, \ldots, c_4$) of the complex Fourier series.

(c) What is the harmonic distortion in the two signals?

(d) Because the harmonic distortion is small, let's concentrate only on the first harmonic. What is the phase shift between input and output signals?

(e) Find the transfer function of the simplest possible linear model that would describe the data. Characterize and interpret the structure of this model. In particular, give a physical explanation for the phase shift.

(f) Predict what the output would be if the model had no phase shift. Would days be hotter? If so, by how much?

4.4 Demodulating an AM Signal

Let $s(t)$ denote a signal that has been amplitude modulated.

$$x(t) = A[1 + s(t)] \sin 2\pi f_c t$$

Radio stations try to restrict the amplitude of the signal $s(t)$ so that it is less than one in magnitude. The frequency f_c is very large compared to the frequency content of the signal. What we are concerned about here is not transmission, but reception.

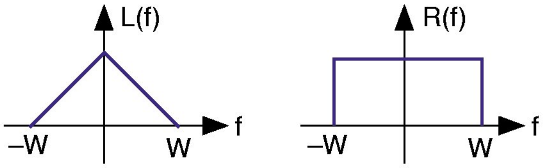
(a) The so-called coherent demodulator simply multiplies the signal $x(t)$ by a sinusoid having the same frequency as the carrier and lowpass filters the result. Analyze this receiver and show that it works. Assume the lowpass filter is ideal.

(b) One issue in coherent reception is the phase of the sinusoid used by the receiver relative to that used by the transmitter. Assuming that the sinusoid of the receiver has a phase ϕ, how does the output depend on ϕ? What is the worst possible value for this phase?

(c) The incoherent receiver is more commonly used because of the phase sensitivity problem inherent in coherent reception. Here, the receiver full-wave rectifies the received signal and lowpass filters the result (again ideally). Analyze this receiver. Does its output differ from that of the coherent receiver in a significant way?

4.5 AM Stereo

A stereophonic signal consists of a "left" signal $l(t)$ and a "right" signal $r(t)$ that conveys sounds coming from an orchestra's left and right sides, respectively. To transmit these two signals simultaneously, the transmitter first forms the sum signal $s_+(t) = l(t) + r(t)$ and the difference signal $s_-(t) = l(t) - r(t)$. Then, the transmitter amplitude-modulates the difference signal with a sinusoid having frequency $2W$, where W is the bandwidth of the left and right signals. The sum signal and the modulated difference signal are added, the sum amplitude-modulated to the radio station's carrier frequency f_c, and transmitted. Assume the spectra of the left and right signals are as shown.



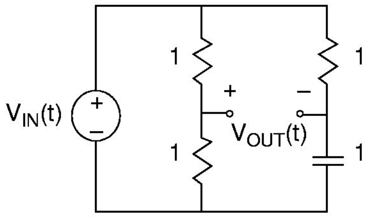
(a) What is the expression for the transmitted signal? Sketch its spectrum.

(b) Show the block diagram of a *stereo* AM receiver that can yield the left and right signals as separate outputs.

(c) What signal would be produced by a conventional coherent AM receiver that expects to receive a standard AM signal conveying a message signal having bandwidth W?

4.6 Phase Distortion

We can learn about phase distortion by returning to circuits and investigate the following circuit.



(a) Find this filter's transfer function.

(b) Find the magnitude and phase of this transfer function. How would you characterize this circuit?

(c) Let $v_{\text{in}}(t)$ be a square-wave of period T. What is the Fourier series for the output voltage?

(d) Use Matlab to find the output's waveform for the cases $T = 0.01$ and $T = 2$. What value of T delineates the two kinds of results you found? The software in `fourier2.m` might be useful.

(e) Now let the phase be linear, amounting to a delay of $T/4$. Use the transfer function of a delay to compute using Matlab the Fourier series of the output. Show that the square wave is indeed delayed.

4.7 Fourier Transform Pairs

Find the Fourier or inverse Fourier transform of the following.

(a) $x(t) = e^{-a|t|}, \; -\infty < t < \infty$ **(b)** $x(t) = te^{-at}\,\mathrm{u}(t)$

(c) $X(f) = \begin{cases} 1, & |f| < W \\ 0, & |f| > W \end{cases}$ **(d)** $x(t) = e^{-at}\cos(2\pi f_0 t)\,\mathrm{u}(t)$

4.8 Lowpass Filtering a Square Wave

Let a square wave (period T) serve as the input to a first-order lowpass system constructed as a RC filter. We want to derive an expression for the time-domain response of the filter to this input.

(a) First, consider the response of the filter to a simple pulse, having unit amplitude and width $T/2$. Derive an expression for the filter's output to this pulse.

(b) Noting that the square wave is a superposition of a sequence of these pulses, what is the filter's response to the square wave?

(c) The nature of this response should change as the relation between the square wave's period and the filter's cutoff frequency change. How long must the period be so that the response does *not* achieve a relatively constant value between transitions in the square wave? What is the relation of the filter's cutoff frequency to the square wave's spectrum in this case?

4.9 Approximating Periodic Signals

Often, we want to approximate a reference signal by a somewhat simpler signal. To assess the quality of an approximation, the most frequently used error measure is the *mean-squared error*. For a periodic signal $s(t)$,

$$\epsilon^2 = \frac{1}{T}\int_0^T [s(t) - \tilde{s}(t)]^2 \, dt$$

where $s(t)$ is the reference signal and $\tilde{s}(t)$ its approximation. One convenient way of finding approximations for periodic signals is to truncate their Fourier series.

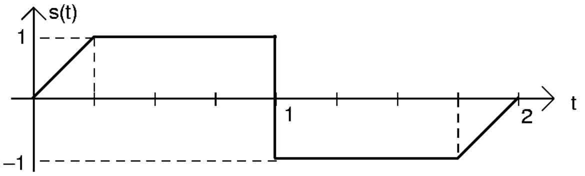
$$\tilde{s}(t) = \sum_{k=-K}^{K} c_k e^{j\frac{2\pi k}{T}t}$$

The point of this problem is to analyze whether this approach is the best (i.e., always minimizes the mean-squared error).

(a) Find an expression for the approximation error when we use the truncated Fourier series as the approximation.

(b) Instead of truncating the series, let's generalize the nature of the approximation to including any set of $2K+1$ terms: We'll always include c_0 and the negative-indexed term corresponding to c_k. What selection of terms minimizes the mean-squared error? Find an expression for the mean-squared error resulting from your choice.

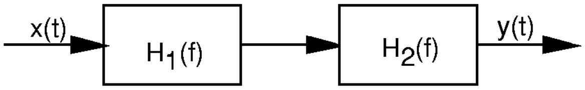
(c) Find the Fourier series for the depicted signal. Use Matlab to find the truncated approximation and best approximation involving two terms. Plot the mean-squared error as a function of K for both approximations.



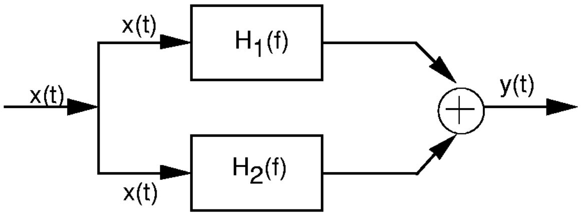
4.10 Arrangements of Systems

Architecting a system of modular components means arranging them in various configurations to achieve some overall input-output relation. For each of the following, determine the overall transfer function between $x(t)$ and $y(t)$.

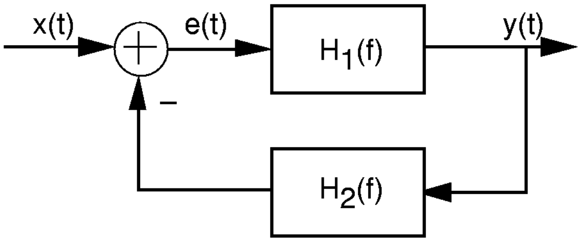
(a)



(b)



(c)



(d) The overall transfer function for part (a) is particularly interesting. What does it say about the effect of the ordering of linear, time-invariant systems in a cascade?

4.11 Reverberation

Reverberation corresponds to adding to a signal its delayed version.

(a) Assuming τ represents the delay, what is the input-output relation for a reverberation system? Is the system linear and time-invariant? If so, find the transfer function; if not, what linearity or time-invariance criterion does reverberation violate.

(b) A music group known as the ROwls is having trouble selling its recordings. The record company's engineer gets the idea of applying different delay to the low and high frequencies and adding the result to create a new musical effect. Thus, the ROwls' audio would be separated into two parts (one less than the frequency f_0, the other greater than f_0), these would be delayed by τ_l and τ_h respectively, and the resulting signals added. Draw a block diagram for this new audio processing system, showing its various components.

(c) How does the magnitude of the system's transfer function depend on the two delays?

4.12 Sampling Signals

If a signal is bandlimited to W Hz, we can sample it at any rate $1/T_s > 2W$ and recover the waveform exactly. This statement of the Sampling Theorem can be taken to mean that all information about the original signal can be extracted from the samples. While true in principle, you do have to be careful how you do so. In addition to the rms value of a signal, an important aspect of a signal is its peak value, which equals $\max |s(t)|$.

(a) Let $s(t)$ be a sinusoid having frequency W Hz. If we sample it at precisely the Nyquist rate, how accurately do the samples convey the sinusoid's amplitude? In other words, find the worst case example.

(b) How fast would you need to sample for the amplitude estimate to be within 5% of the true value?

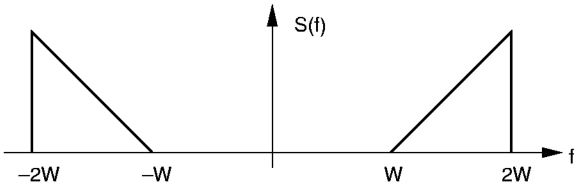
Another issue in sampling is the inherent amplitude quantization produced by A/D converters. Assume the maximum voltage allowed by the converter is $V_{\max}$ volts and that it quantizes amplitudes to b bits.

(c) We can express the quantized sample $Q[s(nT_s)]$ as $s(nT_s)+\epsilon(n)$, where $\epsilon(n)$ represents the quantization error at the n^{th} sample. Assuming the converter rounds, how large is maximum quantization error?

(d) We can describe the quantization error as noise, with a power proportional to the square of the maximum error. What is the signal-to-noise ratio of the quantization error for a full-range sinusoid? Express your result in decibels.

4.13 Bandpass Sampling

The signal $s(t)$ has the indicated spectrum.

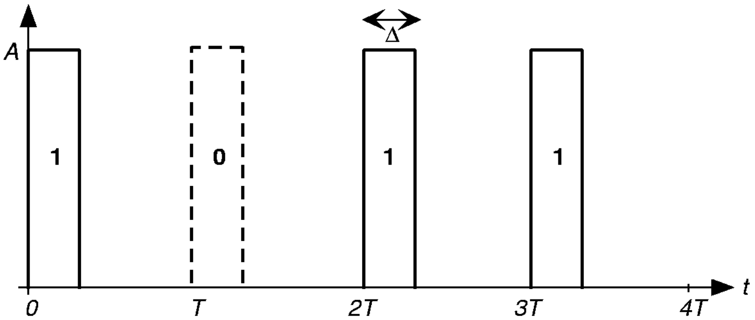


(a) What is the minimum sampling rate for this signal suggested by the Sampling Theorem?

(b) Because of the particular structure of this spectrum, one wonders whether a lower sampling rate could be used. Show that this is indeed the case, and find the system that reconstructs $s(t)$ from its samples.

4.14 Simple D/A Converter

Commercial digital-to-analog converters don't work this way, but this simple circuit illustrates how they work. Let's assume we have a B-bit converter. Thus, we want to convert numbers having a B-bit representation into a voltage proportional to that number. The first step taken by our simple converter is to represent the number by a sequence of B pulses occurring at multiples of a time interval T. The presence of a pulse indicates a "`1`" in the corresponding bit position, and pulse absence means a "`0`" occurred. For a 4-bit converter, the number `13` has the binary representation `1101` ($13_{10} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$) and would be represented by the depicted pulse sequence. Note that the pulse sequence is "backwards" from the binary representation. We'll see why that is.



This signal serves as the input to a first-order RC lowpass filter. We want to design the filter and the parameters Δ and T so that the output voltage at time $4T$ (for a 4-bit converter) is proportional to the number. This combination of pulse creation and filtering constitutes our simple D/A converter. The requirements are

1. The voltage at time $t = 4T$ should diminish by a factor of 2 the further the pulse occurs from this time. In other words, the voltage due to a pulse at $3T$ should be twice that of a pulse produced at $2T$, which in turn is twice that of a pulse at T, etc.
2. The 4-bit D/A converter must support a 10 kHz sampling rate.

Showthe circuit that works. How do the converter's parameters change with sampling rate and number of bits in the converter?

Chapter 5

Digital Signal Processing

In this chapter, we assume that all signals are discrete-time signals, probably but not exclusively derived from analog ones by analog-to-digital conversion. Such signals are *sequences*, a word that means a function defined only for the integers. We thus use the notation $s(n)$ to denote a discrete-time signal. Sequences are fundamentally different than continuous-time signals. For example, continuity has no meaning for a sequence. In this chapter, we discover that digital signal processing is *not* an approximation to analog processing. The theory underlying it —Fourier transforms, linear filtering, and linear systems —parallels what previous chapters described. These similarities make it easy to understand the definitions and why we need them, but the similarities should not be construed as analogies.

The key reason why digital communication systems have a technological advantage today is the *computer*: Computations, like the Fourier transform, can be performed quickly enough to be calculated as the signal is produced,* and programmability means that the signal processing system can be easily changed. This flexibility has an obvious appeal, and has been widely accepted in the marketplace.

5.1 Discrete-Time Signals and Systems

Mathematically, analog signals are functions having as their independent variables continuous quantities, such as space and time. Discrete-time signals are functions defined on the integers; they are sequences. As with analog signals, we seek ways

---

*Taking a systems viewpoint for the moment, a system that produces its output as rapidly as the input arises is said to be a *real-time* system. All analog systems operate in real time; digital ones that depend on a computer to perform system computations may or may not work in real time. Clearly, we need real-time signal processing systems. Only recently have computers become fast enough to meet real-time requirements while performing non-trivial signal processing.

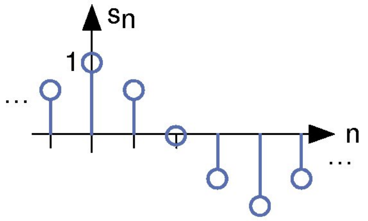


Figure 5.1 The discrete-time cosine signal is plotted as a stem plot. Can you find the formula for this signal?

of decomposing discrete-time signals into simpler components. Because this approach leading to a better understanding of signal structure, we can exploit that structure to represent information (create ways of representing information with signals) and to extract information (retrieve the information thus represented). For symbolic-valued signals, the approach is different: We develop a common representation of all symbolic-valued signals so that we can embody the information they contain in a unified way. From an information representation perspective, the most important issue becomes, for both real-valued and symbolic-valued signals, efficiency; What is the most parsimonious and compact way to represent information so that it can be extracted later.

5.1.1 Real- and complex-valued signals

A discrete-time signal is represented symbolically as $s(n)$, where $n = \dots, -1, 0, 1, \dots$. We usually draw discrete-time signals as stem plots to emphasize the fact they are functions defined only on the integers.

Complex exponentials. The most important signal is, of course, the *complex exponential sequence*.

$$s(n) = e^{j2\pi fn}$$

Sinusoids. Discrete-time sinusoids have the obvious form $s(n) = A\cos(2\pi fn + \phi)$. As opposed to analog complex exponentials and sinusoids that can have their frequencies be any real value, frequencies of their discrete-time counterparts yield unique waveforms *only* when f lies in the interval $\left(-\frac{1}{2}, \frac{1}{2}\right]$. This property can be easily understood by noting that adding an integer to the frequency of the discrete-time complex exponential has no effect on the signal's value.

$$e^{j2\pi(f+m)n} = e^{j2\pi fn}e^{j2\pi mn} = e^{j2\pi fn}$$

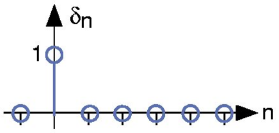
This derivation follows because the complex exponential evaluated at an integer multiple of 2π equals one.

Unit step. Analagous to its analog counterpart, the discrete-time unit step $\mathrm{u}(n)$ sequence is given by

$$\mathrm{u}(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

Unit sample. The second-most important discrete-time signal, the *unit sample* is defined to be

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise.} \end{cases}$$



Examination of a discrete-time signal's plot, like that of the cosine signal shown in Fig. 5.1, reveals that all signals consist of a sequence of delayed and scaled unit samples. Because the value of a sequence at each integer m is denoted by $s(m)$ and the unit sample delayed to occur at m is written $\delta(n-m)$, we can decompose *any* signal as a sum of unit samples delayed to the appropriate location and scaled by the signal value.

$$\boxed{s(n) = \sum_{m=-\infty}^{\infty} s(m)\delta(n-m)}$$

This kind of decomposition is unique to discrete-time signals, and will prove useful subsequently.

5.1.2 Symbolic Signals

Another interesting aspect of discrete-time signals is that their values do not need to be real numbers. We do have real-valued discrete-time signals like the sinusoid, but we also have signals that denote the sequence of characters typed on the keyboard. Such characters certainly aren't real numbers, and as a collection of possible signal values, they have little mathematical structure other than that they are members of a set. Analog signals cannot be symbolic; only in discrete-time can we handle sequences of symbols. More formally, each element of the symbolic-valued signal $s(n)$ takes on one of the values $\{a_1, \ldots, a_K\}$ which comprise the *alphabet* $\mathcal{A}$. This technical terminology does not mean we restrict symbols to being members of the English or Greek alphabet. They could represent keyboard characters, bytes (8-bit quantities), integers that convey daily temperature.

5.2 Digital Processing Systems

Discrete-time systems can act on discrete-time signals in ways similar to those found in analog signals and systems. Because of the role of software in discrete-time systems, many more different systems can be envisioned and "constructed" with programs than can be with analog signals. In fact, a special class of analog signals can be converted into discrete-time signals, processed with software, and converted back into an analog signal, all without the incursion of error. For such signals, systems can be easily produced in software, with equivalent analog realizations difficult, if not impossible, to design. Digital systems are more general than analog systems because they can also deal with symbolic signals.

How do computers perform signal processing? We will discover that digital signals have their own Fourier transform definition and that digital filters can also be designed. We will also discover that digital systems enjoy an *algorithmic* advantage that contributes to rapid processing speeds: Computations can be restructured in non-obvious ways to speed the processing.

Before we explore these ideas, we need to understand what a computer is and how it computes.

- *A simple computer operates fundamentally in discrete time.*
  Computers are *clocked* devices, in which computational steps occur periodically according to ticks of a clock. This description belies clock speed: When you say "I have a 400 MHz computer," you mean that your computer takes 250 nanoseconds to perform each step. That is incredibly fast! A "step" does not, unfortunately, necessarily mean a computation like an addition; computers break such computations down into several stages, which means that the clock speed need not express the computational speed. Computational speed is expressed in units of millions of instructions/second (Mips). Your 400 MHz computer (clock speed) may have a computational speed of 100 Mips.
- *Computers perform integer (discrete-valued) computations.*
  Each computer instruction that performs an elementary calculation —an addition, a multiplication, or a division —does so only for integers. The sum or product of two integers is also an integer, but the quotient of two integers is likely to not be an integer. How does a computer deal with numbers that have digits to the right of the decimal point? This problem is addressed by using the so-called *floating-point* representation of real numbers. At its heart, however, this representation relies on integer-valued computations.

We need to concentrate on how computers represent numbers and perform arithmetic. Suppose we have twenty-five "things." This number is an abstract quantity that we need to represent so that we can perform calculations on it. We represent this number with *positional notation* using base 10: $25_{10} = 2\times10^1+5\times10^0 =$ twenty-five. We could also use base-8 notation ($31_8 = 3 \times 8^1 + 1 \times 8^0$), base 16 ($1\text{F} = 1 \times 16^1 + \text{F} \times 16^0$, with F representing fifteen), or base 2 ($11001_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$). Each of these represent the *same* quantity; all numbers can be so represented and arithmetic can be performed in each system. Which we chose is a matter of convenience and cultural heritage. Each position is occupied by an integer between 0 and the base minus 1, and the integer's position implicitly represents how many times the base raised to an exponent is needed. The integer N is succinctly expressed in base-b as $N_b = d_n d_{n-1} \ldots d_0$, which mathematically means $N_b = d_n b^n + \cdots + d_0 b^0$. No matter what base you might choose, addition, subtraction, multiplication, and division all follow the same rules you learned as a child. These rules can be derived mathematically from the positional notation conventions defined by this formula. * To extend the positional representation to signed integers, we add a symbol that represents whether the number is positive or negative.

Humans use base 10, the origin of which is commonly assumed to be the fact we have ten fingers. Digital computers use the base 2 or *binary* number representation, each digit of which is known as a *bit* (*b*inary dig*it*). Here, each bit is represented as a voltage that is either "high" or "low," thereby representing "1" or "0", respectively. To represent signed values, we tack on a special bit —the *sign bit* —to express the sign. The binary addition and multiplication tables are

$$\begin{array}{llll} 0+0=0 & 1+1=10 & 0\cdot0=0 & 1\cdot1=1 \\ 0+1=1 & 1+0=1 & 0\cdot1=0 & 0\cdot1=0 \end{array}$$

Note that if carries are ignored,* subtraction of two single-digit binary numbers yields the same bit as addition. Computers use high and low voltage values to express a bit, and an array of such voltages express numbers akin to positional notation. Logic circuits perform arithmetic operations.

Question 5.1 {**254**} *Add twenty-five and seven in base 2. Note the carries that might occur. Why is the result "nice"?*

Also note that the logical operations of AND and OR are equivalent to binary addition (again if carries are ignored). The variables of logic indicate truth or falsehood. $A \cap B$, the AND of A and B, represents a statement that both A and B must be true for the statement to be true. You use this kind of statement to tell search engines that you want to restrict hits to cases where both of the events A and B occur. $A \cup B$, the OR of A and B, yields a value of truth if either is true. Note that if we represent truth by a "1" and falsehood by a "0", *binary multiplication corresponds to* AND *and addition (ignoring carries) to* OR. The Irish mathematician George Boole discovered this equivalence in the mid-nineteenth century. It laid the foundation for what we now call Boolean algebra, which expresses as equations logical statements. More importantly, any computer using base-2 representations and arithmetic can also easily evaluate logical statements. This fact makes an integer-based computational device much more powerful than might be apparent.

---

*Non-positional systems are very strange; take Roman numerals for example. Try adding, or more extremely dividing, two numbers using this number representation system.

*A *carry* means that a computation performed at a given position affects other positions as well. Here, $1 + 1 = 10$ is an example of a computation that involves a carry.

Computers express numbers in a fixed-size collection of bits, commonly known as the computer's *word length*. Today, word-lengths are either 32 or 64 bits, corresponding to a power-of-two number of bytes (8-bit "chunks"). This design choice restricts the largest integer (in magnitude) that can be represented on a computer.

Question 5.2 $\{$**254**$\}$ *For both 32-bit and 64-bit integer representations, what is the largest number that can be represented? Don't forget that the sign bit must also be included.*

When it comes to expressing fractions, positional notation is easily extended to negative exponents using the decimal point convention: Wherever the decimal point occurs in a string of digits, we know that the first digit to the left corresponds to the zero exponent, and the one to the right is -1. In this way, we have $2.5_{10} = 2 \times 10^0 + 5 \times 10^{-1}$. *Computers don't use this convention to represent numbers with fractional parts*; instead, they use a generalization of scientific notation. Here, a number (not necessarily an integer) x is expressed as $m \times b^e$, with the *mantissa* m lying within a proscribed range (scientific notation requires $1 \leq |m| < 10$), b is the base, and e the integer *exponent*. Computers use a convention known as *floating point*, where base 2 is used and the mantissa must lie between one-half and one. Thus, two and one-half in the computer's number representation scheme becomes five-eighths times four: $.101 \times 2^2$. The mantissa is stored in most of the bits in a computer's word, and the exponent stored in the remaining bits. Both the exponent and mantissa have sign bits associated with them. Thirty-two-bit floating point numbers, known as single-precision floating point, have eight bits representing the exponent (including the sign bit) and the remaining twenty-four representing the mantissa (again including a sign bit).

Question 5.3 $\{$**254**$\}$ *What are the largest and smallest numbers that can be represented in 32-bit floating point? 64-bit floating point that has sixteen bits allocated to the exponent? Note that both exponent and mantissa require a sign bit.*

So long as the integers aren't too large, they can be represented exactly in a computer using the binary positional notation. Electronic circuits that make up the physical computer can add and subtract integers without error.* Floating point representation handles numbers with fractional parts, but only some with no error. Similar to the integer case, the number could be too big or too small to be so represented. More fundamentally, *many* numbers cannot be accurately represented no matter how many bits are used for the exponent and mantissa. For example, you know that the fraction 1/3 has an infinite decimal (base 10) expansion; no matter how many decimal digits you have in your calculator, you will never represent 1/3 with complete accuracy. It also has an infinite binary expansion as well.

Question 5.4 {**254**} *Can 1/3 be expressed in a finite number of digits in any base? If so, what bases do the job?*

Clearly, many numbers have infinite expansions, and this situation applies to binary expansions as well. Consequently, number representation and arithmetic performed by a computer cannot be infinitely accurate in most cases. The errors incurred in most calculations will be small, but this fundamental source of error can cause trouble at times.

5.3 Discrete-Time Fourier Transform (DTFT)

Now that we have the underpinnings of digital computation, we need to return to signal processing ideas. The most prominent of which is, of course, the Fourier transform. The Fourier transform of a sequence is defined to be

$$S(e^{j2\pi f}) = \sum_{-\infty}^{\infty} s(n) e^{-j2\pi f n} \tag{5.1}$$

Frequency here has no units. As should be expected, this definition is linear, with the transform of a sum of signals equaling the sum of their transforms. Real-valued signals have conjugate-symmetric spectra: $S(e^{-j2\pi f}) = \left[S(e^{j2\pi f})\right]^*$.

Question 5.5 {**254**} *A special property of the discrete-time Fourier transform is that it is periodic with period one: $S(e^{j2\pi(f+1)}) = S(e^{j2\pi f})$. Derive this property from the definition of the DTFT.*

Because of this periodicity, we need only plot the spectrum over one period to understand completely the spectrum's structure; typically, we plot the spectrum over the frequency range $[-\frac{1}{2}, \frac{1}{2}]$. When the signal is real-valued, we can further simplify our plotting chores by showing the spectrum only over $[0, \frac{1}{2}]$; the spectrum at negative frequencies can be derived from positive-frequency spectral values.

---

*This statement isn't quite true; when does addition cause problems?

When we obtain the discrete-time signal via sampling an analog signal, the Nyquist frequency corresponds to the discrete-time frequency $\frac{1}{2}$. To show this, note that a sinusoid at the Nyquist frequency $1/2T_s$ has a sampled waveform that equals

$$\cos\left(2\pi \cdot \frac{1}{2T_s} \cdot nT_s\right) = \cos \pi n = (-1)^n .$$

The exponential in the DTFT at frequency $\frac{1}{2}$ equals $e^{-j2\pi n/2} = e^{-j\pi n} = (-1)^n$, meaning that the correspondence between analog and discrete-time frequency is established:

$$\boxed{f_D = f_A T_s} ,$$

where f_D and f_A represent discrete-time and analog frequency variables, respectively. Figure 4.17 $\{120\}$ provides another way of deriving this result. As the duration of each pulse in the periodic sampling signal $p_{T_s}(t)$ narrows, the amplitudes of the signal's spectral repetitions, which are governed by the Fourier series coefficients of $p_{T_s}(t)$ (Eq. (4.18) $\{119\}$), become increasingly equal.* Thus, the sampled signal's spectrum becomes period with period $1/T_s$. Thus, the Nyquist frequency $1/2T_s$ corresponds to the frequency $1/2$.

Example

Let's compute the discrete-time Fourier transform of the exponentially decaying sequence $s(n) = a^n\, u(n)$, where $u(n)$ is the unit-step sequence. Simply plugging the signal's expression into the Fourier transform formula,

$$\begin{aligned} S(e^{j2\pi f}) &= \sum_{n=-\infty}^{\infty} a^n\, u(n) e^{-j2\pi f n} \\ &= \sum_{n=0}^{\infty} \left(a e^{-j2\pi f}\right)^n \end{aligned}$$

---

*Examination of Eq. (4.18) $\{119\}$ reveals that as Δ decreases, the value of c_0, the largest Fourier coefficient, decreases to zero: $|c_0| = A\Delta/T$. Thus, to maintain a mathematically viable Sampling Theorem, the amplitude A must increase as $1/\Delta$, becoming infinitely large as the pulse duration decreases. Practical systems use a small value of Δ, say $0.1 \cdot T_s$ and use amplifiers to rescale the signal.

This sum is a special case of the *geometric series*.

$$\boxed{\sum_{n=0}^{\infty} \alpha^n = \frac{1}{1-\alpha},\ |\alpha| < 1} \tag{5.2}$$

Thus, as long as $|a| < 1$, we have our Fourier transform.

$$S(e^{j2\pi f}) = \frac{1}{1 - ae^{-j2\pi f}}$$

Using Euler's relation, we can express the magnitude and phase of this spectrum.

$$|S(e^{j2\pi f})| = \frac{1}{\sqrt{(1 - a\cos 2\pi f)^2 + a^2 \sin^2 2\pi f}}$$

$$\angle S(e^{j2\pi f}) = -\tan^{-1} \frac{a \sin 2\pi f}{1 - a\cos 2\pi f}$$

No matter what value of a we chose, the above formulae clearly demonstrate the periodic nature of the spectra of discrete-time signals. Fig. 5.2 shows indeed that the spectrum is a periodic function. We need only consider the spectrum between $-\frac{1}{2}$ and $\frac{1}{2}$ to unambiguously define it. When $a > 0$, we have a lowpass spectrum—the spectrum diminishes as frequency increases from 0 to $\frac{1}{2}$—with increasing a leading to a greater low frequency content; for $a < 0$, we have a highpass spectrum (Fig. 5.3).

Example

Analogous to the analog pulse signal, let's find the spectrum of the length-N pulse sequence.

$$s(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases}$$

The Fourier transform of this sequence has the form of a truncated geometric series.

$$S(e^{j2\pi f}) = \sum_{n=0}^{N-1} e^{-j2\pi f n}$$

For the so-called finite geometric series, we know that

$$\boxed{\sum_{n=n_0}^{N+n_0-1} \alpha^n = \alpha^{n_0} \frac{1-\alpha^N}{1-\alpha}} \tag{5.3}$$

for *all* values of α.

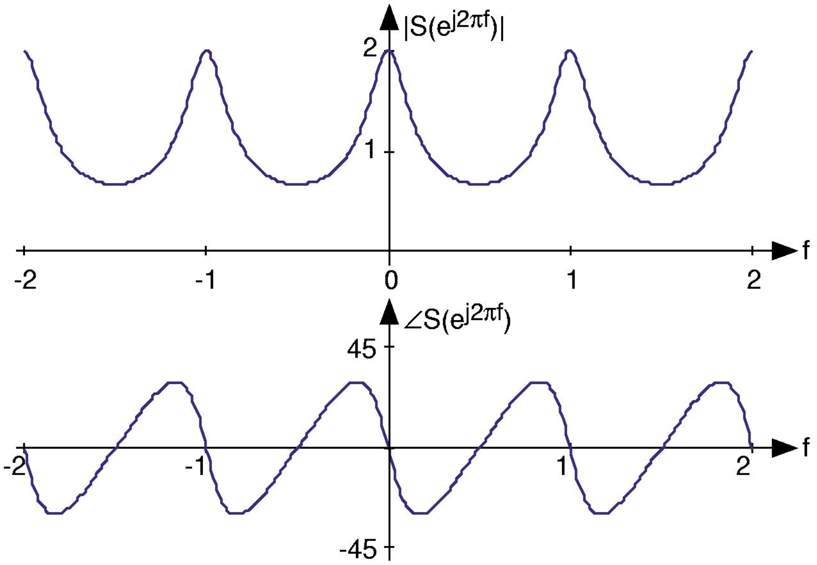


Figure 5.2 The spectrum of the exponential signal ($a = 0.5$) is shown over the frequency range $[-2, 2]$, clearly demonstrating the periodicity of all discrete-time spectra. The angle has units of degrees.

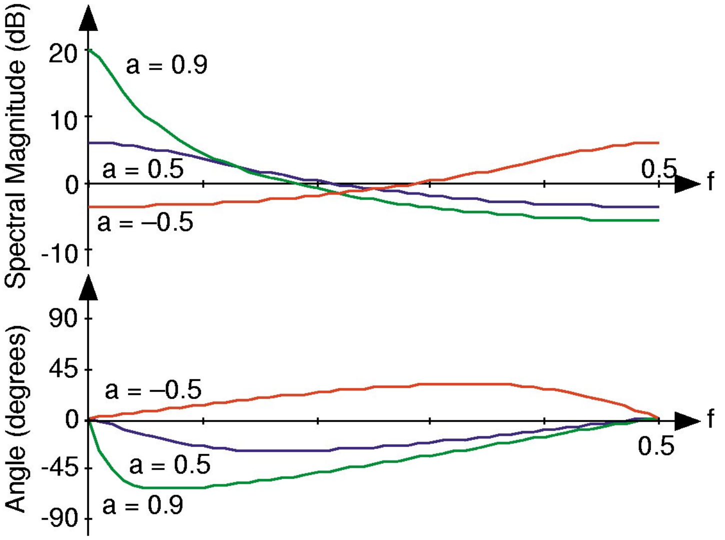


Figure 5.3 The spectra of several exponential signals are shown. What is the apparent relationship between the spectra for $a = 0.5$ and $a = -0.5$?

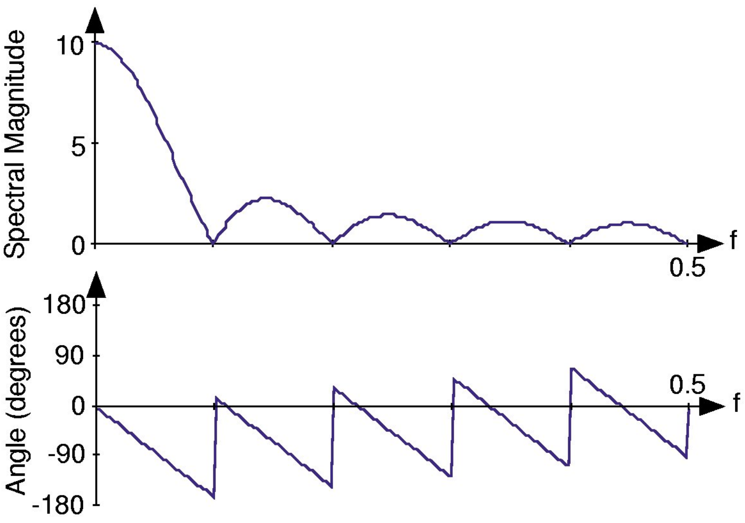


Figure 5.4 The spectrum of a length-ten pulse is shown. Can you explain the rather complicated appearance of the phase?

Question 5.6 {**255**} *Derive this formula for the finite geometric series sum. The "trick" is to consider the difference between the series' sum and the sum of the series multiplied by* α*.*

Applying this result yields (Fig. 5.4)

$$\begin{aligned} S(e^{j2\pi f}) &= \frac{1-e^{-j2\pi fN}}{1-e^{-j2\pi f}} \\ &= e^{-j\pi f(N-1)}\frac{\sin \pi fN}{\sin \pi f} \end{aligned}$$

The ratio of sine functions has the generic form of $\sin Nx / \sin x$, which is known as the *discrete-time sinc function* $\operatorname{dsinc} x$. Thus, our transform can be concisely expressed as $S(e^{j2\pi f}) = e^{-j\pi f(N-1)} \operatorname{dsinc} \pi f$. The discrete-time pulse's spectrum contains many ripples, the number of which increase with N, the pulse's duration.

The inverse discrete-time Fourier transform is easily derived from the following relationship.

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-j2\pi fm} e^{+j2\pi fn}\, df = \begin{cases} 1, & m=n \\ 0, & m \neq n \end{cases}$$

Therefore, we find that

$$\begin{aligned}\int_{-\frac{1}{2}}^{\frac{1}{2}} S(e^{j2\pi f})e^{+j2\pi fn}\,df &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \sum_m s(m)e^{-j2\pi fm}e^{+j2\pi fn}\,df \\ &= \sum_m s(m)\int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-j2\pi f(m-n)}\,df \\ &= s(n)\end{aligned}$$

The Fourier transform pairs in discrete-time are

$$\boxed{\begin{aligned} S(e^{j2\pi f}) &= \sum_n s(n)e^{-j2\pi fn} \\ s(n) &= \int_{-\frac{1}{2}}^{\frac{1}{2}} S(e^{j2\pi f})e^{+j2\pi fn}\,df \end{aligned}} \tag{5.4}$$

The properties of the discrete-time Fourier transform mirror those of the analog Fourier transform. Table 5.1 shows similarities and differences. One important common property is Parseval's Theorem.

$$\boxed{\sum_{n=-\infty}^{\infty} |s(n)|^2 = \int_{-\frac{1}{2}}^{\frac{1}{2}} |S(e^{j2\pi f})|^2\,df}$$

To show this important property, we simply substitute the Fourier transform expression into the frequency-domain expression for power.

$$\begin{aligned}\int_{-\frac{1}{2}}^{\frac{1}{2}} |S(e^{j2\pi f})|^2\,df &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\sum_n s(n)e^{-j2\pi fn}\right)\left(\sum_m s^*(n)e^{+j2\pi fm}\right)df \\ &= \sum_{n,m} s(n)s^*(m)\int_{-\frac{1}{2}}^{\frac{1}{2}} e^{+j2\pi f(m-n)}\,df\end{aligned}$$

Using the orthogonality relation demonstrated previously {141}, the integral equals $\delta(m-n)$, where $\delta(n)$ is the unit sample {133}. Thus, the double sum collapses into a single sum because nonzero values occur only when $n = m$, giving Parseval's Theorem as a result. We term $\sum_n s(n)^2$ the energy in the discrete-time signal $s(n)$ in spite of the fact that discrete-time signals don't consume (or produce for that matter) energy. This terminology is a carry-over from the analog world.

Discrete-Time Fourier Transform Properties

	Sequence-Domain	Frequency Domain
Linearity	$a_1 s_1(n) + a_2 s_2(n)$	$a_1 S_1(e^{j2\pi f}) + a_2 S_2(e^{j2\pi f})$
Conjugate Symmetry	$s(n)$ real	$S(e^{j2\pi f}) = S^*(e^{-j2\pi f})$
Even Symmetry	$s(n) = s(-n)$	$S(e^{j2\pi f}) = S(e^{-j2\pi f})$
Odd Symmetry	$s(n) = -s(-n)$	$S(e^{j2\pi f}) = -S(e^{-j2\pi f})$
Time Delay	$s(n - n_0)$	$e^{-j2\pi f n_0} S(e^{j2\pi f})$
Complex Modulation	$e^{j2\pi f_0 n} s(n)$	$S(e^{j2\pi(f-f_0)})$
Amplitude Modulation	$s(n)\ \cos 2\pi f_0 n$	$\dfrac{S(e^{j2\pi(f-f_0)}) + S(e^{j2\pi(f+f_0)})}{2}$
	$s(n)\ \sin 2\pi f_0 n$	$\dfrac{S(e^{j2\pi(f-f_0)}) - S(e^{j2\pi(f+f_0)})}{2j}$
Multiplication by n	$ns(n)$	$\dfrac{1}{-j2\pi}\dfrac{dS(e^{j2\pi f})}{df}$
Sum	$\displaystyle\sum_{n=-\infty}^{\infty} s(n)$	$S(e^{j2\pi\cdot 0})$
Value at origin	$s(0)$	$\displaystyle\int_{-\frac{1}{2}}^{\frac{1}{2}} S(e^{j2\pi f})\, df$
Parseval's Theorem	$\displaystyle\sum_{n=-\infty}^{\infty} \lvert s(n)\rvert^2$	$\displaystyle\int_{-\frac{1}{2}}^{\frac{1}{2}} \left\lvert S(e^{j2\pi f})\right\rvert^2 df$

Table 5.1 Discrete-time Fourier transform properties and relations.

Question 5.7 {**255**} *Suppose we obtained our discrete-time signal from values of the product* $s(t)\,\mathrm{p}_{T_s}(t)$, *where the duration of the component pulses in* $\mathrm{p}_{T_s}(t)$ *is* Δ. *How is the discrete-time signal energy related to the total energy contained in* $s(t)$? *Assume the signal is bandlimited and that the sampling rate was chosen appropriate to the Sampling Theorem's conditions.*

5.4 Discrete Fourier Transform (DFT)

The discrete-time Fourier transform (and the continuous-time transform as well) can be evaluated when we have an analytic expression for the signal. Suppose we just have a signal, such as the speech signal used in the previous chapter. You might be curious; how did we compute the spectrogram shown in Figure 4.15 $\{118\}$? The big difference between the continuous-time and discrete-time worlds is that we can *exactly* calculate spectra in discrete-time. For analog-signal spectra, use must build special devices, which turn out in most cases to consist of A/D converters and discrete-time computations. Certainly discrete-time spectral analysis is more flexible than in continuous-time.

The formula for the DTFT (5.1) $\{137\}$ is a sum, which conceptually can be easily computed save for two issues.

- *Signal duration.*
  The sum extends over the signal's duration, which must be finite to compute the signal's spectrum. It is exceedingly difficult to store an infinite-length signal in any case, so we'll assume that the signal extends over $[0, N-1]$.
- *Continuous frequency.*
  Subtler than the signal duration issue is the fact that the frequency variable is continuous: It may only need to span one period, like $[-\frac{1}{2}, \frac{1}{2}]$ or $[0, 1]$, but the DTFT formula as it stands requires evaluating the spectra at *all* frequencies within a period. Let's compute the spectrum at a few frequencies; the most obvious ones are the equally spaced ones $f = k/K$, $k = 0, \dots, K-1$.

We thus define the *discrete Fourier transform* (DFT) to be

$$\boxed{S(k) = \sum_{n=0}^{N-1} s(n) e^{-j\frac{2\pi nk}{K}} \qquad k = 0, \dots, K-1} \tag{5.5}$$

Here, $S(k)$ is shorthand for $S(e^{j2\pi k/K})$.

We can compute the spectrum at as many equally spaced frequencies as we like. Note that you can think about this computationally motivated choice as *sampling* the spectrum; more about this interpretation later. The issue now is how many frequencies are enough to capture how the spectrum changes with frequency. One way of answering this question is determining an inverse discrete Fourier transform formula: given $S(k)$, $k = 0, \dots, K-1$, how do we find $s(n)$, $n = 0, \dots, N-1$? Presumably, the formula will be of the form $s(n) \overset{?}{=} \sum_{k=0}^{K-1} S(k) e^{+j2\pi nk/K}$. Substituting the DFT formula in this prototype inverse transform yields

$$s(n) \overset{?}{=} \sum_{k=0}^{K-1} \sum_{m=0}^{N-1} s(m) e^{-j\frac{2\pi mk}{K}} e^{+j\frac{2\pi nk}{K}}$$

Note that the orthogonality relation we use so often has a different character now.

$$\sum_{k=0}^{K-1} e^{-j\frac{2\pi km}{K}} e^{+j\frac{2\pi kn}{K}} = \begin{cases} K, & m = n, n \pm K, n \pm 2K, \ldots \\ 0, & \text{otherwise} \end{cases}$$

We obtain nonzero value whenever the two indices differ by multiples of K. We can express this result as $K \sum_l \delta(m - n - lK)$. Thus, our formula becomes

$$s(n) \overset{?}{=} \sum_{m=0}^{N-1} s(m) K \sum_{l=-\infty}^{\infty} \delta(m - (n + lK)) \ .$$

The integers n and m both range over $0, \ldots, N-1$. To have an inverse transform, we need the sum to be a *single* unit sample for m, n in this range. If it did not, then $s(n)$ would equal a sum of values, and we would not have a valid transform: Once going into the frequency domain, we could not get back unambiguously! Clearly, the term $l = 0$ always provides a unit sample (we'll take care of the factor of K soon). If we evaluate the spectrum at *fewer* frequencies than the signal's duration, the term corresponding to $m = n + K$ will also appear for some values of $m, n = 0, \ldots, N-1$. This situation means that our prototype transform equals $s(n) + s(n+K)$ for some values of n. The only way to eliminate this problem is to require $K \geq N$: *We must have more frequency samples than the signal's duration.* In this way, we can return from the frequency domain we entered via the DFT.

Question 5.8 {**255**} *When we have fewer frequency samples than the signal's duration, some discrete-time signal values equal the sum of the original signal values. Given the sampling interpretation of the spectrum, characterize this effect a different way.*

Another way to understand this requirement is to use the theory of linear equations. If we write out the expression for the DFT as a set of linear equations,

$$\begin{aligned} s(0) + s(1) + \cdots + s(N-1) &= S(0) \\ s(0) + s(1)e^{-j\frac{2\pi}{K}} + \cdots + s(N-1)e^{-j\frac{2\pi(N-1)}{K}} &= S(1) \\ &\vdots \\ s(0) + s(1)e^{-j\frac{2\pi(K-1)}{K}} + \cdots + s(N-1)e^{-j\frac{2\pi(N-1)(K-1)}{K}} &= S(K-1) \end{aligned}$$

we have K equations in N unknowns if we want to find the signal from its sampled spectrum. This requirement is impossible to fulfill if $K < N$; we must have $K \geq N$. Our orthogonality relation essentially says that if we have a sufficient number of equations (frequency samples), the resulting set of equations can indeed be solved.

By convention, the number of DFT frequency values K is chosen to equal the signal's duration N. The discrete Fourier transform pair consists of

$$\boxed{S(k) = \sum_{n=0}^{N-1} s(n) e^{-j\frac{2\pi nk}{N}} \qquad s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) e^{+j\frac{2\pi nk}{N}}} \tag{5.6}$$

5.4.1 Computational Complexity

We now have a way of computing the spectrum for an arbitrary signal: The discrete Fourier transform (DFT) (5.5) computes the spectrum at N equally spaced frequencies from a length-N sequence. An issue that never arises in analog "computation," like that performed by a circuit, is how much work it takes to perform the signal processing operation such as filtering. In computation, this consideration translates to the number of basic computational steps required to perform the needed processing. The number of steps, known as the *complexity*, becomes equivalent to how long the computation takes (how long must we wait for an answer). Complexity is not so much tied to specific computers or programming languages but to how many steps are required on any computer. Thus, a procedure's stated complexity says that the time taken will be *proportional* to some function of the amount of data used in the computation and the amount demanded.

For example, consider the formula for the discrete Fourier transform. For each frequency we chose, we must multiply each signal value by a complex number and add together the results. For a real-valued signal, each real-times-complex multiplication requires two real multiplications, meaning we have $2N$ multiplications to perform. To add the results together, we must keep the real and imaginary parts separate. Adding N numbers requires $N - 1$ additions. Consequently, each frequency requires $2N + 2(N - 1) = 4N - 2$ basic computational steps. As we have N frequencies, the total number of computations is $N(4N - 2)$.

In complexity calculations, we only worry about what happens as the data lengths increase, and take the dominant term —here the $4N^2$ term—as reflecting how much work is involved in making the computation. As multiplicative constants don't matter since we are making a "proportional to" evaluation, we find the DFT is an $O\left(N^2\right)$ computational procedure. This notation is read "order N-squared". Thus, if we double the length of the data, we would expect that the computation time to approximately quadruple.

Question 5.9 {**255**} *In making the complexity evaluation for the DFT, we assumed the data to be real. Three questions emerge. First of all, the spectra of such signals have conjugate symmetry, meaning that negative frequency components ($k = N/2 + 1, \dots, N - 1$ in Eq. 5.5) can be computed from the corresponding positive frequency components. Does this symmetry change the DFT's complexity?*

Secondly, suppose the data are complex-valued; what is the DFT's complexity now?

Finally, a less important but interesting question is suppose we want K frequency values instead of N; now what is the complexity?

5.4.2 The Fast Fourier Transform

One wonders if the DFT can be computed faster: Does another computational procedure—an *algorithm*—exist that can compute the same quantity, but more efficiently. We could seek methods that reduce the constant of proportionality, but do not change the complexity from $O\left(()\, N^2\right)$. Here, we have something more dramatic in mind: Can the computations be restructured so that a *smaller* complexity results?

In 1965, IBM researcher Jim Cooley and Princeton faculty member John Tukey developed what is now known as the fast Fourier transform (FFT). It is an algorithm for computing that DFT that has order $O\left(N \log N\right)$ *for certain length inputs*. Now when the length of data doubles, the spectral computational time will not quadruple as with the DFT algorithm; instead, it approximately doubles. Later research showed that no algorithm for computing the DFT could have a smaller complexity than the FFT. Surprisingly, historical work has shown that Gauss in the early nineteenth century developed the same algorithm, but did not publish it! After the FFT's rediscovery, not only was the computation of a signal's spectrum greatly speeded, but also the added feature of *algorithm* meant that computations had flexibility not available to analog implementations.

Question 5.10 {**255**} *Before developing the FFT, let's try to appreciate the algorithm's impact. Suppose a short-length transform takes 1 ms. We want to calculate a transform of a signal that is 10 times longer. Compare how much longer a straightforward implementation of the DFT would take in comparison to an FFT, both of which compute exactly the same quantity.*

To derive the FFT, we assume that the signal's duration is a power of two: $N = 2^\ell$. Consider what happens to the even-numbered and odd-numbered elements of the sequence in the DFT calculation.

$$\begin{aligned}
S(k) &= s(0) + s(2)e^{-j\frac{2\pi 2k}{N}} + \cdots + s(N-2)e^{-j\frac{2\pi(N-2)k}{N}} \\
&\quad + s(1)e^{-j\frac{2\pi k}{N}} + s(3)e^{-j\frac{2\pi(2+1)k}{N}} + \cdots + s(N-1)e^{-j\frac{2\pi(N-2+1)k}{N}} \\
&= \left[s(0) + s(2)e^{-j\frac{2\pi k}{N/2}} + \cdots + s(N-2)e^{-j\frac{2\pi(N/2-1)k}{N/2}}\right] \\
&\quad + \left[s(1) + s(3)e^{-j\frac{2\pi k}{N/2}} + \cdots + s(N-1)e^{-j\frac{2\pi(N/2-1)k}{N/2}}\right] e^{-j2\pi k/N}
\end{aligned} \tag{5.7}$$

Each term in square brackets has the *form* of a $N/2$-length DFT. The first one is a DFT of the even-numbered elements, and the second of the odd-numbered elements. The first DFT is combined with the second multiplied by the complex exponential $e^{-j2\pi k/N}$. The half-length transforms are each evaluated at frequency indices $k = 0, \ldots, N-1$. Normally, the number of frequency indices in a DFT calculation range between zero and the transform length minus one. *The computational advantage of the FFT comes from recognizing the periodic nature of the discrete Fourier transform.* The FFT simply reuses the computations made in the half-length transforms and combines them through additions and the multiplication by $e^{-j2\pi k/N}$, which is not periodic over $N/2$, to rewrite the length-N DFT. Figure 5.5 illustrates this decomposition. As it stands, we now compute two length-$N/2$ transforms (complexity $2O\left(N^2/4\right)$), multiply one of them by the complex exponential (complexity $O\left(N\right)$), and add the results (complexity $O\left(N\right)$). At this point, the total complexity is still dominated by the half-length DFT calculations, but the proportionality coefficient has been reduced.

Now for the fun. Because $N = 2^\ell$, each of the half-length transforms can be reduced to two quarter-length transforms, each of these to two eighth-length ones, etc. This decomposition continues until we are left with length-2 transforms. This transform is quite simple, involving only additions. Thus, the first stage of the FFT has $N/2$ length-2 transforms (see the bottom part of Fig. 5.5). Pairs of these transforms are combined by adding one to the other multiplied by a complex exponential. Each pair requires 4 additions and 4 multiplications, giving a total number of computations equaling $8 \cdot N/4 = N/2$. This number of computations does not change from stage to stage. Because the number of stages, the number of times the length can be divided by two, equals $\log_2(N)$, the complexity of the FFT is $O\left(N \log N\right)$.

By going through an example, how computational savings are achieved becomes more evident. Let's look at the details of a length-8 DFT. As shown on Fig. 5.5, we first decompose the DFT into two length-4 DFTs, with the outputs

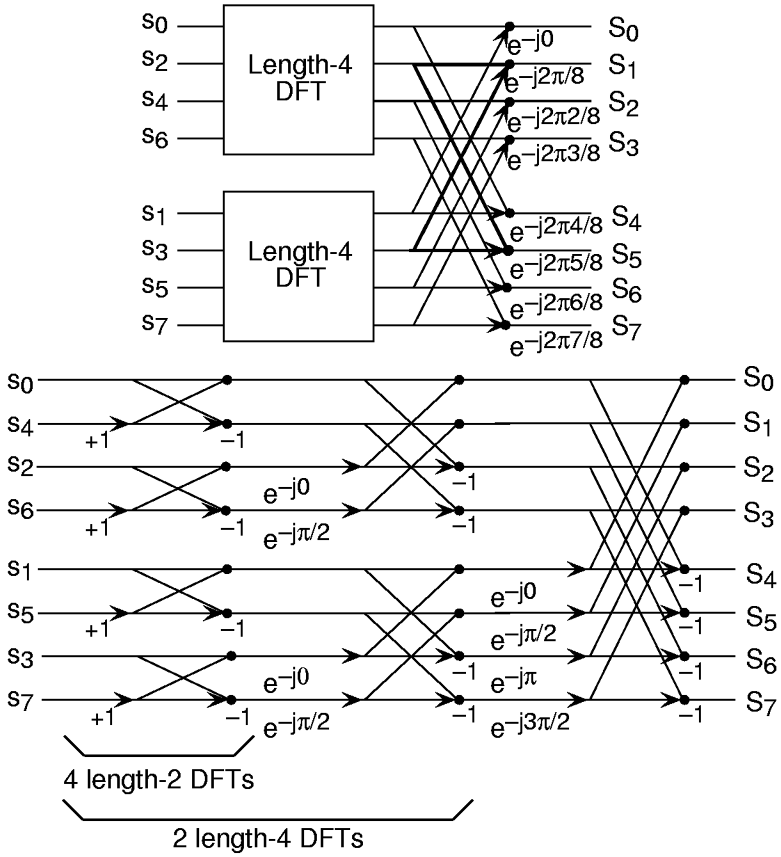

s0
s2
s4
s6
Length-4 DFT
s1
s3
s5
s7
Length-4 DFT
e−j0
e−j2π/8
e−j2π2/8
e−j2π3/8
e−j2π4/8
e−j2π5/8
e−j2π6/8
e−j2π7/8
S0
S1
S2
S3
S4
S5
S6
S7
s0
s4
s2
s6
s1
s5
s3
s7
+1
−1
e−j0
e−jπ/2
e−jπ
e−j3π/2
4 length-2 DFTs
2 length-4 DFTs

Figure 5.5 The initial decomposition of a length-8 DFT into the terms using even- and odd-indexed inputs marks the first phase of developing the FFT algorithm. When these half-length transforms are successively decomposed, we are left with the diagram shown in the bottom panel that depicts the length-8 FFT computation.

added and subtracted together in pairs. Considering Eq. (5.7) as the frequency index goes from 0 through 7, we recycle values from the length-4 DFTs into the final calculation because of the periodicity of the DFT output. Examining how pairs of outputs are collected together, we create the basic computational element known as a *butterfly* (Fig. 5.6). By considering together the computations involving common output frequencies from the two half-length DFTs, we see that the two complex multiplies are related to each other, and we can reduce our computational work even further. By further decomposing the length-4 DFTs into two length-2 DFTs and combining their outputs, we arrive at the diagram summarizing the length-8

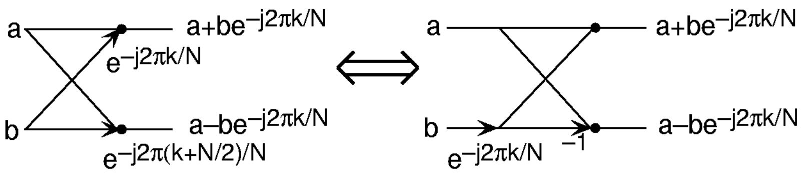


Figure 5.6 The basic computational element of the fast Fourier transform is the butterfly. It takes two complex numbers, represented by a and b, and forms the quantities shown. Each butterfly requires one complex multiplication and two complex additions.

fast Fourier transform (Fig. 5.5). Although most of the complex multiplies are quite simple (multiplying by $e^{-j\pi}$ means negating real and imaginary parts), let's count those for purposes of evaluating the complexity as full complex multiplies. We have $N/2 = 4$ complex multiplies and $2N = 16$ additions for each stage and $\log_2 N = 3$ stages, making the number of basic computations $\frac{3N}{2} \log_2 N$ as predicted.

Question 5.11 $\{$**255**$\}$ *Note that the ordering of the input sequence in the two parts of Fig. 5.5 aren't quite the same. Why not? How is the ordering determined?*

Other "fast" algorithms were discovered, all of which make use of how many common factors the transform length N has. In number theory, the number of prime factors a given integer has measures how *composite* it is. The numbers 16 and 27 are highly composite (equaling 2^4 and 3^4 respectively), the number 18 is less so ($2^2 \cdot 3^2$), and 17 not at all (it's prime). To get some glimpse into relative computational efficiencies, look at Problem 5.5 $\{174\}$. In over thirty years of Fourier transform algorithm development, the original Cooley-Tukey algorithm is far and away the most frequently used. It is so computationally efficient that power-of-two transform lengths are frequently used regardless of what the actual length of the data.

5.4.3 Spectrograms

Now that we know how to acquire analog signals for digital processing (pre-filtering, sampling, and A/D conversion) and to compute spectra of discrete-time signals (using the FFT algorithm), let's put these various components together to learn how the spectrogram shown in Figure 4.15 $\{118\}$ was calculated. The speech was sampled at a rate of 11.025 kHz* and passed through a 16-bit A/D converter.

---

*Music compact discs (CDs) encode their signals at a sampling rate of 44.1 kHz. We'll learn the rationale for this number later. The 11.025 kHz sampling rate for the speech is 1/4 of the CD sampling rate, and was the lowest available sampling rate commensurate with speech signal bandwidths available on my computer.

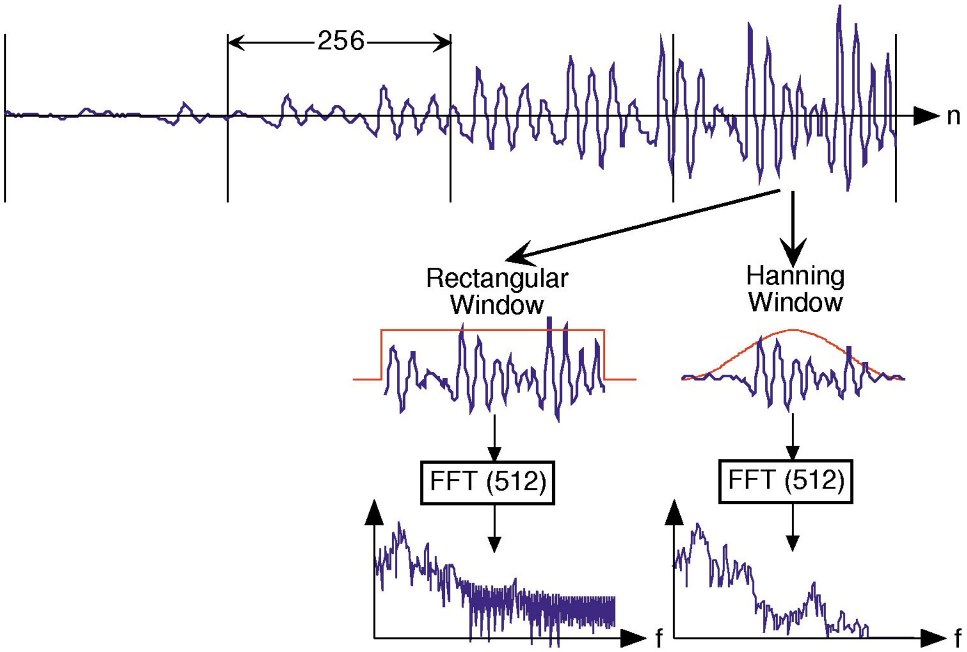


Figure 5.7 The top waveform is a segment 1024 samples long taken from the beginning of the "Rice University" phrase. Computation the spectrogram shown in Figure 4.15 {118} involved creating frames, here demarked by the vertical lines, that were 256 samples long and finding the spectrum of each. If a rectangular window is applied (corresponding to extracting a frame from the signal), oscillations appear in the spectrum (middle of bottom row). Applying a Hanning window gracefully tapers the signal toward frame edges, thereby yielding a more accurate computation of the signal's spectrum at that moment of time.

Question 5.12 {**255**} *Looking at Figure 4.15 {118}, the signal lasted a little over 1.2 seconds. How long was the sampled signal (in terms of samples)? What was the datarate during the sampling process in bps (bits per second)? Assuming the computer storage is organized in terms of bytes (8-bit quantities), how many bytes of computer memory does the speech consume?*

The resulting discrete-time signal, shown in the bottom of Fig. 4.15, clearly changes its character with time. To display these spectral changes, the long signal was *sectioned* into *frames*: comparatively short, contiguous groups of samples. Conceptually, a Fourier transform of each frame is calculated using the FFT. Each frame is not so long that significant signal variations are retained within a frame, but not so short that we lose the signal's spectral character.

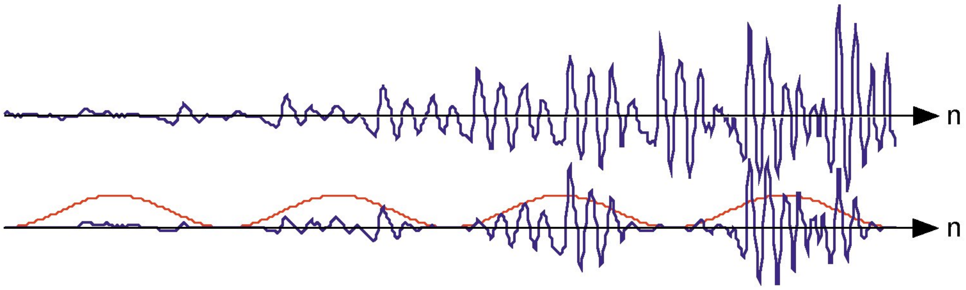


Figure 5.8 In comparison with the original speech segment shown in the upper plot, the non-overlapped Hanning windowed version shown below it is very ragged. Clearly, spectral information extracted from the bottom plot could well miss important features present in the original.

An important detail emerges when we examine each framed signal (Fig. 5.7). At the frame's edges, the signal may change very abruptly, a feature not present in the original signal. A transform of such a segment reveals a curious oscillation in the spectrum, an artifact directly related to this sharp amplitude change. A better way to frame signals for spectrograms is to apply a *window*: Shape the signal values within a frame so that the signal decays gracefully as it nears the edges. This shaping is accomplished by multiplying the framed signal by the sequence $w(n)$. In sectioning the signal, we essentially applied a rectangular window: $w(n) = 1$, $n \leq 0 < N$. A much more graceful window is the *Hanning window*; it has the cosine shape $w(n) = \frac{1}{2}(1 - \cos 2\pi n/N)$. As shown in Fig. 5.7, this shaping greatly reduces spurious oscillations in each frame's spectrum. Considering the spectrum of the Hanning windowed frame, we find that the oscillations resulting from applying the rectangular window obscured a formant (the one located at a little more than half the Nyquist frequency).

Question 5.13 $\{$**256**$\}$ *What might be the source of these oscillations? To gain some insight, what is the length-*$2N$ *discrete Fourier transform of a length-*N *pulse? The pulse emulates the rectangular window, and certainly has edges. Compare your answer with the length-*$2N$ *transform of a length-*N *Hanning window.*

If you examine the windowed signal sections in sequence to examine windowing's affect on signal amplitude, we see that we have managed to amplitude-modulate the signal with the periodically repeated window (Fig. 5.8). To alleviate this problem, frames are overlapped (typically by half a frame duration). This solution requires more Fourier transform calculations than needed by rectangular windowing, but the spectra are much better behaved and spectral changes are much better captured.

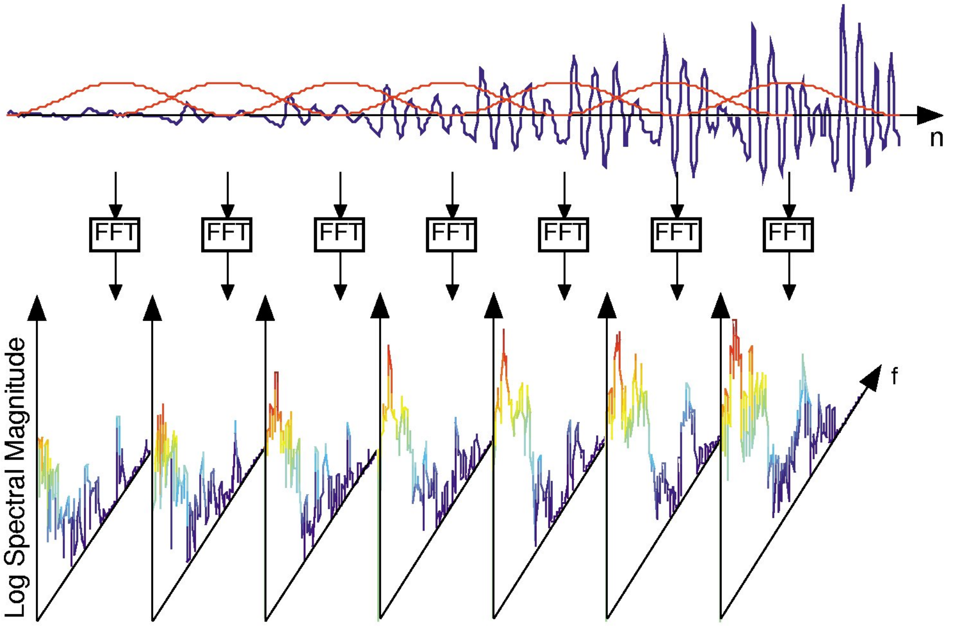


Figure 5.9 The original speech segment and the sequence of overlapping Hanning windows applied to it are shown in the upper portion. Frames were 256 samples long and a Hanning window was applied with a half-frame overlap. A length-512 FFT of each frame was computed, with the magnitude of the first 257 FFT values displayed vertically, with spectral amplitude values color-coded.

Question 5.15 {256}

Spectrograms, such as shown in Figure 4.15 {118}, are sectioned into overlapping, equal-length frames, with a Hanning window applied to each frame. The spectra of each of these is calculated, and displayed in spectrograms with frequency extending vertically, window time location running horizontally, and spectral magnitude color-coded. Fig. 5.9 illustrates these computations.

Question 5.14 {256} *Why the specific values of 256 for N and 512 for K? Another issue is how was the length-512 transform of each length-256 windowed frame computed?*

5.5 Discrete-Time Systems

When we developed analog systems, interconnecting the circuit elements provided a natural starting place for constructing useful devices. In discrete-time signal processing, we are not limited by hardware considerations but by what can be constructed in software.

One of the first analog systems we described was the amplifier {16}. We found that implementing an amplifier was difficult in analog systems, requiring an op-amp at least. What is the discrete-time implementation of an amplifier? Is this especially hard or easy?

In fact, we will discover that frequency-domain implementation of systems, wherein we multiply the input signal's Fourier transform by a frequency response, is not only a viable alternative, but also a computationally efficient one. We begin with discussing the underlying mathematical structure of linear, shift-invariant systems, and devise how software filters can be constructed.

5.5.1 Systems in the Time-Domain

A discrete-time signal $s(n)$ is *delayed* by n_0 samples when we write $s(n-n_0)$, with $n_0 > 0$. Choosing n_0 to be negative advances the signal along the integers. As opposed to analog delays {16}, discrete-time delays can *only* be integer valued. In the frequency domain, delaying a signal corresponds to a linear phase shift of the signal's discrete-time Fourier transform: $s(n-n_0) \leftrightarrow e^{-j2\pi f n_0} S(e^{j2\pi f})$.

Linear discrete-time systems have the superposition property.

$$\mathcal{S}\left[a_1 x_1(n) + a_2 x_2(n)\right] = a_1 \mathcal{S}\left[x_1(n)\right] + a_2 \mathcal{S}\left[x_2(n)\right]$$

A discrete-time system is called *shift-invariant* (analogous to time-invariant analog systems {18}) if delaying the input delays the corresponding output.

$$\text{If } \mathcal{S}\left[x(n)\right] = y(n), \text{ then } \mathcal{S}\left[x(n-n_0)\right] = y(n-n_0)$$

We use the term shift-invariant to emphasize that delays can only have integer values in discrete-time, while in analog signals, delays can be arbitrarily valued.

We want to concentrate on systems that are both linear and shift-invariant. It will be these that allow us the full power of frequency-domain analysis and implementations. Because we have no physical constraints in "constructing" such systems, we need only a mathematical specification. In analog systems, the differential equation specifies the input-output relationship in the time-domain. The corresponding discrete-time specification is the *difference equation*.

$$\boxed{y(n) = a_1 y(n-1) + \cdots + a_p y(n-p) + b_0 x(n) + b_1 x(n-1) + \cdots + b_q x(n-q)} \tag{5.8}$$

Here, the output signal $y(n)$ is related to its *past* values $y(n-l)$, $l = 1, \ldots, p$, and to the current and past values of the input signal $x(n)$. The system's characteristics are determined by the choices for the number of coefficients p and q and the coefficients' values $\{a_1, \dots, a_p\}$ and $\{b_0, b_1, \dots, b_q\}$.*

As opposed to differential equations, which only provide an *implicit* description of a system (we must somehow solve the differential equation), difference equations provide an *explicit* way of computing the output for any input. We simply express the difference equation by a program that calculates each output from the previous output values, and the current and previous inputs.

Example

Let's consider the simple system having $p = 1$ and $q = 0$.

$$y(n) = ay(n-1) + bx(n) \tag{5.9}$$

To compute the output at some index, this difference equation says we need to know what the previous output $y(n-1)$ and what the input signal is at that moment of time. In more detail, let's compute this system's output to a unit-sample input: $x(n) = \delta(n)$. Because the input is zero for negative indices, we start by trying to compute the output at $n = 0$.

$$y(0) = ay(-1) + b$$

What is the value of $y(-1)$? Because we have used an input that is zero for all negative indices, it is reasonable to assume that the output is also zero. Certainly, the difference equation would not describe a linear system if the input that is zero for *all* time did not produce a zero output $\{18\}$. With this assumption, $y(-1) = 0$, leaving $y(0) = b$. For $n > 0$, the input unit-sample is zero, which leaves us with the difference equation $y(n) = ay(n-1)$, $n > 0$. We can envision how the filter responds to this input by making a table.

$$y(n) = ay(n-1) + b\delta(n)$$

n	$x(n)$	$y(n)$
-1	0	0
0	b	b
1	0	$b \cdot a$
2	0	$b \cdot a^2$
$\vdots$	0	$\vdots$
n	0	$b \cdot a^n$

Coefficient values determine how the output behaves. The parameter b can be any value, and serves as a gain. The effect of the parameter a is more complicated

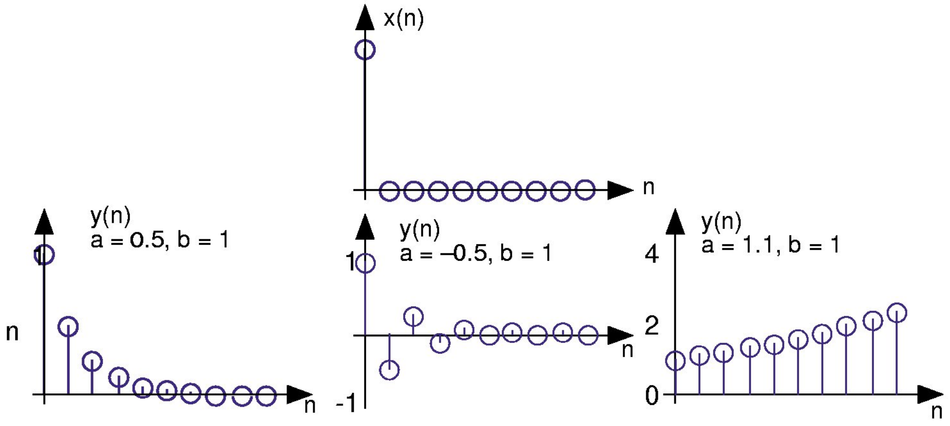


Figure 5.10 The input to the simple example system, a unit sample, is shown at the top, with the outputs for several system parameter values shown below.

(Fig. 5.10). If it equals zero, the output simply equals the input times the gain b. For all non-zero values of a, the output lasts forever; such systems are said to be *IIR* (*I*nfinite *I*mpulse *R*esponse). The reason for this terminology is that the unit sample also known as the impulse (especially in analog situations), and the system's response to the "impulse" lasts forever. If a is positive and less than one, the output is a decaying exponential. When $a = 1$, the output is a unit step. If a is negative and greater than -1, the output oscillates while decaying exponentially. When $a = -1$, the output changes sign forever, alternating between b and $-b$. More dramatic effects when $|a| > 1$; whether positive or negative, the output signal becomes larger and larger, *growing* exponentially.

Positive values of a are used in population models to describe how population size increases over time. Here, n might correspond to generation. The difference equation says that the number in the next generation is some multiple of the previous one. If this multiple is less than one, the population becomes extinct; if greater than one, the population flourishes. The same difference equation also describes the effect of compound interest on deposits. Here, n indexes the times at which compounding occurs (daily, monthly, etc.), a equals the compound interest rate plus one, and $b = 1$ (the bank provides no gain). In signal processing applications, we typically require that the output remain bounded for any input. For our example, that means that we restrict $|a| < 1$ and chose values for it and the

---

[*] There is an asymmetry in the coefficients: where is a_0? This coefficient would multiply the $y(n)$ term in Eq. 5.8. We have essentially divided the equation by it, which does not change the input-output relationship. We have thus created the convention that a_0 is always one.

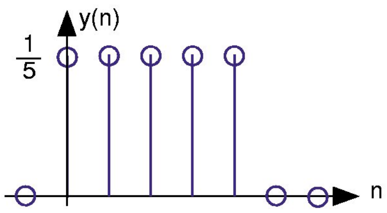


Figure 5.11 The plot shows the unit-sample response of a length-5 boxcar filter.

gain according to the application.

Question 5.16 {**256**} *Note that the difference equation* (5.8) {*154*} *does not involve terms like* $y(n+1)$ *or* $x(n+1)$ *on the equation's right side. Can such terms also be included? Why or why not?*

Example

A somewhat different system has no "*a*" coefficients. Consider the difference equation

$$y(n) = \frac{1}{q}\left[x(n) + \cdots + x(n-q+1)\right] . \tag{5.10}$$

Because this system's output depends only on current and previous input values, we need not be concerned with initial conditions. When the input is a unit-sample, the output equals $1/q$ for $n = 0, \ldots, q-1$, then equals zero thereafter. Such systems are said to be *FIR* (*F*inite *I*mpulse *R*esponse) because their unit sample responses have finite duration. Plotting this response (Fig. 5.11) shows that the unit-sample response is a pulse of width q and height $1/q$. This waveform is also known as a boxcar, hence the name "boxcar filter" given to this system. (We'll derive its frequency response and develop its filtering interpretation in the next section.) For now, note that the difference equation says that each output value equals the *average* of the input's current and previous values. Thus, the output equals the running average of input's previous q values. Such a system could be used to produce the average weekly temperature ($q = 7$) that could be updated daily.

Difference equations are usually expressed in software with `for` loops. A MATLAB program that would compute the first 1000 values of the output has the form

```
for n=1:1000
  y(n) = sum(a.*y(n-1:-1:n-p)) + sum(b.*x(n:-1:n-q));
end
```

An important detail emerges when we consider making this program work; in fact, as written it has (at least) two bugs. What input and output values enter into the computation of `y(1)`? We need values for $y(0), y(-1), \ldots$, values we have not yet computed. To compute them, we would need more previous values of the output, which we have not yet computed. To compute these values, we would need even earlier values, *ad infinitum.* The way out of this predicament is to specify the system's *initial conditions*: we must provide the p output values that occurred before the input started. These values can be arbitrary, but the choice does impact how the system responds to a given input. *One* choice gives rise to a linear system: Make the initial conditions zero. The reason lies in the definition of a linear system given previously {154}: The only way that the output to a sum of signals can be the sum of the individual outputs occurs when the initial conditions in each case are zero.

Question 5.17 {256} *The initial condition issue resolves making sense of the difference equation for inputs that start at some index. However, the program will not work because of a programming, not conceptual, error. What is it? How can it be "fixed?"*

5.5.2 Systems in the Frequency Domain

As with analog linear systems, we need to find the frequency response of discrete-time systems. We used impedances to derive directly from the circuit's structure the frequency response. The only structure we have so far for a discrete-time system is the difference equation. We proceed as when we used impedances: let the input be a complex exponential signal. When we have a linear, shift-invariant system, the output should also be a complex exponential of the same frequency, changed in amplitude and phase. These amplitude and phase changes comprise the frequency response we seek.

The complex exponential input signal is $x(n) = Xe^{j2\pi fn}$.* Assume the output has a similar form: $y(n) = Ye^{j2\pi fn}$. Plugging these signals into the fundamental difference equation (5.8) {154}, we have

$$\begin{aligned} Ye^{j2\pi fn} &\overset{?}{=} a_1Ye^{j2\pi f(n-1)} + \cdots + a_pYe^{j2\pi f(n-p)} \\ &\quad + b_0Xe^{j2\pi fn} + b_1Xe^{j2\pi f(n-1)} + \cdots + b_qXe^{j2\pi f(n-q)} \ . \end{aligned}$$

---

*Note that this input occurs for *all* values of n. No need to worry about initial conditions here.

The assumed output does indeed satisfy the difference equation if the output complex amplitude is related to the input amplitude by

$$Y = \frac{b_0 + b_1 e^{-j2\pi f} + \cdots + b_q e^{-j2\pi q f}}{1 - a_1 e^{-j2\pi f} - \cdots - a_p e^{-j2\pi p f}} X \ .$$

This relationship corresponds to the system's frequency response or, by another name, its transfer function. We find that any discrete-time system defined by a difference equation has a transfer function given by

$$\boxed{H(e^{j2\pi f}) = \frac{b_0 + b_1 e^{-j2\pi f} + \cdots + b_q e^{-j2\pi q f}}{1 - a_1 e^{-j2\pi f} - \cdots - a_p e^{-j2\pi p f}}} \tag{5.11}$$

Furthermore, because *any* discrete-time signal can be expressed as a superposition of complex exponential signals and because linear discrete-time systems obey the Superposition Principle, the transfer function relates the discrete-time Fourier transform of the system's output to the input's Fourier transform.

$$\boxed{Y(e^{j2\pi f}) = H(e^{j2\pi f}) X(e^{j2\pi f})} \tag{5.12}$$

Example

The frequency response of the simple IIR system (difference equation given by (5.9) {155}) is given by

$$H(e^{j2\pi f}) = \frac{b}{1 - a e^{-j2\pi f}} \ .$$

This Fourier transform occurred in a previous example; Figure 5.2 {140} portrays the magnitude and phase of this transfer function. When the filter coefficient a is positive, we have a lowpass filter; negative a results in a highpass filter. The larger the coefficient in magnitude, the more pronounced the low- or highpass filtering.

Example

The length-q boxcar filter (difference equation found in (5.10) {157}) has the frequency response

$$H(e^{j2\pi f}) = \frac{1}{q} \sum_{m=0}^{q-1} e^{-j2\pi f m} \ .$$

This expression amounts to the Fourier transform of the boxcar signal, which was analyzed previously ({139}). There we found that this frequency response has a magnitude equal to the absolute value of $\text{dsinc}\,\pi f$; see Figure 5.4 {141} for the length-10 filter's frequency response. We see that boxcar filters —length-q signal averagers — have a lowpass behavior, having a cutoff frequency of $1/q$.

Question 5.18 {**256**} *Suppose we multiply the boxcar filter's coefficients by a sinusoid:* $b_m = \frac{1}{q}\cos 2\pi f_0 m$. *Use Fourier transform properties to determine the transfer function. How would you characterize this system: Does it act like a filter? If so, what kind of filter and how do you control its characteristics with the filter's coefficients?*

These examples illustrate the point that systems described (and implemented) by difference equations serve as filters for discrete-time signals. The filter's *order* is given by the number p of denominator coefficients in the transfer function (if the system is IIR) or by the number q of numerator coefficients if the filter is FIR. When a system's transfer function has both terms, the system is usually IIR, and its order equals p regardless of q. By selecting the coefficients and filter type, filters having virtually any frequency response desired can be designed. This design flexibility can't be found in analog systems. In the next section, we detail how analog signals can be filtered by computers, offering a much greater range of filtering possibilities than is possible with circuits.

5.6 Discrete-Time Filtering of Analog Signals

Because of the Sampling Theorem, we can process, in particular filter, analog signals "with a computer" by constructing the system shown in Fig. 5.12. To use this system, we are assuming that the input signal has a lowpass spectrum and can be bandlimited without affecting important signal aspects.* Note that the input and output filters must be analog filters; trying to operate without them can lead to potentially very inaccurate digitization.

Another implicit assumption is that the digital filter can operate in *real time*: The computer and the filtering algorithm must be sufficiently fast so that outputs are computed faster than input values arrive. The sampling interval, which is determined by the analog signal's bandwidth, thus determines how long our program has to compute *each* output $y(n)$. The computational complexity for calculating each output with a difference equation (5.8) {154} is $O(p+q)$. Frequency domain implementation of the filter is also possible. The idea begins by computing the Fourier transform of a length-N portion of the input $x(n)$, multiplying it by the

---

*Bandpass signals can also be filtered digitally, but require a more complicated system. Highpass signals cannot be filtered digitally.

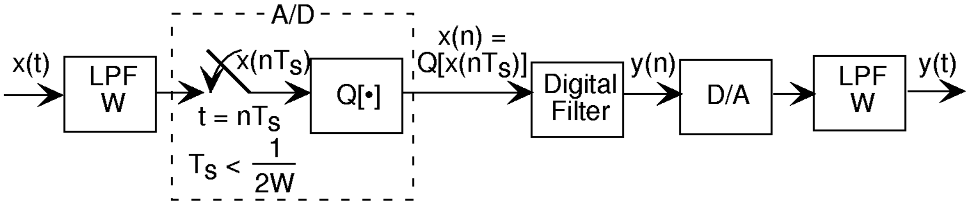


Figure 5.12 To process an analog signal digitally, the signal $x(t)$ must be filtered with an anti-aliasing filter (to ensure a bandlimited signal) before A/D conversion. This lowpass filter (LPF) has a cutoff frequency of W Hz, which determines allowable sampling intervals T_s. The greater the number of bits in the amplitude quantization portion $Q[\cdot]$ of the A/D converter, the greater the accuracy of the entire system. The resulting digital signal $x(n)$ can now be filtered in the time-domain with a difference equation or in the frequency domain with Fourier transforms. The resulting output $y(n)$ then drives a D/A converter and a second anti-aliasing filter (having the same bandwidth as the first one).

filter's transfer function, and computing the inverse transform of the result. This approach seems overly complex and potentially inefficient. Detailing the complexity, however, we have $O(N \log N)$ for the two transforms (computed using the FFT algorithm) and $O(N)$ for the multiplication by the transfer function, which makes the total complexity $O(N \log N)$ *for N input values.* A frequency domain implementation thus requires $O(\log N)$ computational complexity for each output value. The complexities of time-domain and frequency-domain implementations depend on different aspects of the filtering: The time-domain implementation depends on the combined orders of the filter while the frequency-domain implementation depends on the logarithm of the Fourier transform's length.

It could well be that in some problems the time-domain version is more efficient (more easily satisfies the real time requirement) and in others the frequency domain approach is faster. In the latter situations, it is the FFT algorithm for computing the Fourier transforms that enables the potential superiority of frequency-domain approaches. Because complexity considerations only express how algorithm running-time increases with system parameter choices, we need to detail both implementations to determine which will be more suitable for any given filtering problem. Filtering with a difference equation is straightforward, and the number of computations that must be made for each output value is $2(p+q)$.

Question 5.19 $\{256\}$ *Derive the exact number of computations for the general difference equation* (5.8) $\{$*154*$\}$.

As for the frequency domain ...

Filtering in the Frequency Domain

Because we are interested in actual computations rather than analytic calculations, we must consider the details of the discrete Fourier transform. To compute the length-N DFT, we assume that the signal has a duration less than or equal to N. Because frequency responses have an explicit frequency-domain specification in terms of filter coefficients (Eq. (5.11) $\{159\}$), we don't have a direct handle on which signal has a Fourier transform equaling a given frequency response. Finding this signal is quite easy. First of all, note that the discrete-time Fourier transform of a unit sample equals one for all frequencies. Because of the input and output of linear, shift-invariant systems are related to each other by $Y(e^{j2\pi f}) = H(e^{j2\pi f})X(e^{j2\pi f})$, *a unit-sample input results in the output's Fourier transform equaling the system's transfer function.*

Question 5.20 $\{$**256**$\}$ *This statement is a very important result. Derive it yourself.*

In the time-domain, the output for a unit-sample input is known as the system's *unit-sample response*, and is denoted by $h(n)$. Combining the frequency-domain and time-domain interpretations of a linear, shift-invariant system's unit-sample response, we have that $h(n)$ and the transfer function are Fourier transform pairs *in terms of the discrete-time Fourier transform.*

$$\boxed{h(n) \longleftrightarrow H(e^{j2\pi f})} \tag{5.13}$$

Returning to the issue of how to use the DFT to perform filtering, we can analytically specify the frequency response, and derive the corresponding length-N DFT by sampling the frequency response.

$$H(k) = H(e^{j2\pi k/N}), \quad k = 0, \ldots, N-1$$

Computing the inverse DFT yields a length-N signal *no matter what the actual duration of the unit-sample response might be.* If the unit-sample response has a duration less than or equal to N (it's a FIR filter), computing the inverse DFT of the sampled frequency response indeed yields the unit-sample response. If, however, the duration exceeds N, errors are encountered. The nature of these errors is easily explained by appealing to the Sampling Theorem. By sampling in the frequency domain, we have the potential for aliasing in the time domain (sampling in one domain, be it time or frequency, can result in aliasing in the other) unless we sample fast enough. Here, the duration of the unit-sample response determines the minimal sampling rate that prevents aliasing. For FIR systems —they by definition have finite-duration unit sample responses —the number of required DFT samples equals the unit-sample response's duration: $N \geq q$.

Question 5.21 {**256**} *Derive the minimal DFT length for a length-q unit-sample response using the Sampling Theorem. Because sampling in the frequency domain causes repetitions of the unit-sample response in the time domain, sketch the time-domain result for various choices of the DFT length N.*

Question 5.22 {**257**} *Express the unit-sample response of a FIR filter in terms of difference equation coefficients. Note that the corresponding question for IIR filters is far more difficult to answer: Consider the example on page* {*155*}.

For IIR systems, we cannot use the DFT to find the system's unit-sample response: aliasing of the unit-sample response will *always* occur. Consequently, we can only implement an IIR filter accurately in the time domain with the system's difference equation. Frequency-domain implementations are restricted to FIR filters.

Another issue arises in frequency-domain filtering that is related to time-domain aliasing, this time when we consider the output. Assume we have an input signal having duration N_x that we pass through a FIR filter having a length-$q+1$ unit-sample response. What is the duration of the output signal? The difference equation for this filter is

$$y(n) = b_0 x(n) + \ldots b_q x(n-q) \ .$$

This equation says that the output depends on current and past input values, with the input value q samples previous defining the extent of the filter's *memory* of past input values. For example, the output at index N_x depends on $x(N_x)$ (which equals zero), $x(N_x - 1)$, through $x(N_x - q)$. Thus, the output returns to zero only after the last input value passes through the filter's memory. As the input signal's last value occurs at index $N_x - 1$, the last nonzero output value occurs at when $n - q = N_x - 1$ or $n = q + N_x - 1$. Thus, the output signal's duration equals $q + N_x$.

Question 5.23 {**257**} *In words, we express this result as "The output's duration equals the input's duration plus the filter's duration minus one." Demonstrate the accuracy of this statement.*

The main theme of this result is that a filter's output extends longer than either its input or its unit-sample response. Thus, to avoid aliasing when we use DFTs, the dominant factor is not the duration of input or of the unit-sample response, but of the output. Thus, the number of values at which we must evaluate the frequency response's DFT must be at least $q + N_x$ and we must compute the same length DFT of the input. To accommodate a shorter signal than DFT length, we simply *zero-pad* the input: Ensure that for indices extending beyond the signal's duration that the signal is zero. Frequency-domain filtering, diagrammed in

Fig. 5.13, is accomplished by storing the filter's frequency response as the DFT $H(k)$, computing the input's DFT $X(k)$, multiplying them to create the output's DFT $Y(k) = H(k) \cdot X(k)$, and computing the inverse DFT of the result to yield $y(n)$.

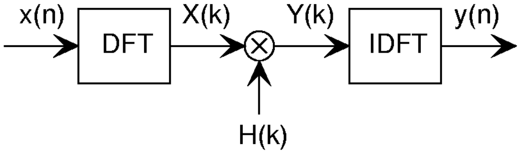


Figure 5.13 To filter a signal in the frequency domain, first compute the DFT of the input, multiply the result by the sampled frequency response, and finally compute the inverse DFT of the product. The DFT's length *must* be at least the sum of the input's and unit-sample response's duration minus one. We calculate these discrete Fourier transforms using the fast Fourier transform algorithm, of course.

Before detailing this procedure, let's clarify why so many new issues arose in trying to develop a frequency-domain implementation of linear filtering. The previously derived frequency-domain relationship between a filter's input and output (Eq. (5.12) {159}) is *always* true: $Y(e^{j2\pi f}) = H(e^{j2\pi f})X(e^{j2\pi f})$. This Fourier transforms in this result are discrete-time Fourier transforms; for example, $X(e^{j2\pi f}) = \sum_n x(n)e^{-j2\pi fn}$. Unfortunately, using this relationship to perform filtering is restricted to the situation when we have analytic formulas for the frequency response and the input signal. The reason why we had to "invent" the discrete Fourier transform (DFT) has the same origin: The spectrum resulting from the discrete-time Fourier transform depends on the *continuous* frequency variable f. That's fine for analytic calculation, but computationally we would have to make an uncountably infinite number of computations.* The DFT computes the Fourier transform at a finite set of frequencies—samples the true spectrum—which can lead to aliasing in the time-domain unless we sample sufficiently fast. The sampling interval here is $1/K$ for a length-K DFT: faster sampling to avoid aliasing thus requires a longer transform calculation. Since the longest signal among the input, unit-sample response and output is the output, it is that signal's duration that determines the transform length. We simply extend the other two signals with zeros (zero-pad) to compute their DFTs.

---

*Did you know that two kinds of infinities can be meaningfully defined? A *countably infinite* quantity means that it can be associated with a limiting process associated with integers. An *uncountably infinite* quantity cannot be so associated. The number of rational numbers is countably infinite (the numerator and denominator correspond to locating the rational by row and column; the total number so-located can be counted, voila!); the number of irrational numbers is uncountably infinite. Guess which is "bigger?"

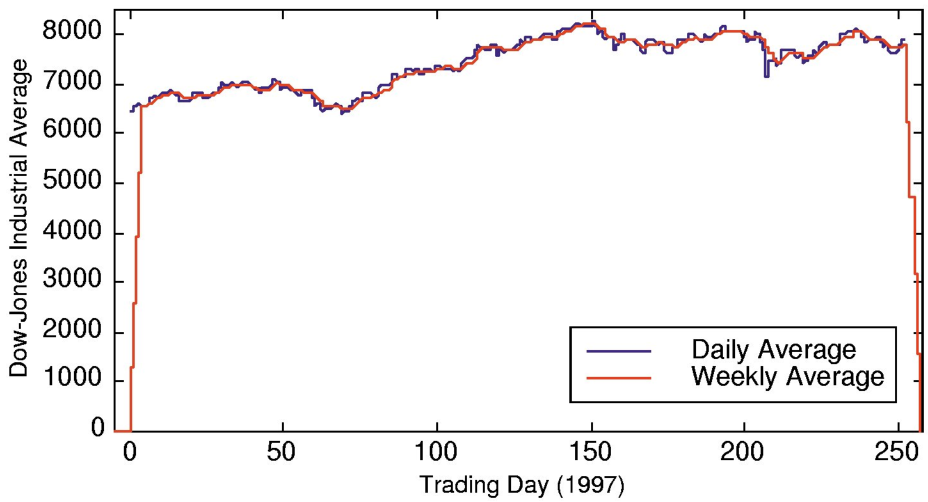


Figure 5.14 The blue line shows the Dow Jones Industrial Average from 1997, and the red one the length-5 boxcar-filtered result that provides a running weekly of this market index. Note the "edge" effects in the filtered output.

Example

Suppose we want to average daily stock prices taken over last year to yield a running weekly average (average over five trading sessions). The filter we want is a length-5 averager (the unit-sample response of which is shown in Figure 5.11 {157}), and the input's duration is 253 (365 calendar days minus weekend days and holidays). The output duration will be $253 + 5 - 1 = 257$, and this determines the transform length we need to use. Because we want to use the FFT, we are restricted to power-of-two transform lengths. We need to choose any FFT length that exceeds the required DFT length. As it turns out, 256 is a power of two ($2^8 = 256$), and this length just undershoots our required length. To use frequency domain techniques, we must use length-512 fast Fourier transforms.

Fig. 5.14 shows the input and the filtered output. The MATLAB programs that compute the filtered output in the time and frequency domains are

Time Domain

```
h = [1 1 1 1 1]/5;
y = filter(h,1,[djia zeros(1,4)]);*
```

Frequency Domain

```
h = [1 1 1 1 1]/5;
DJIA = fft(djia, 512);
H = fft(h, 512);
Y = H.*X;
y = ifft(Y);
```

MATLAB's `fft` function automatically zero-pads its input if the specified transform length (its second argument) exceeds the signal's length. The frequency domain result will have a small imaginary component —largest value is 2.2×10^{-11} —because of the inherent finite precision nature of computer arithmetic. Because of the unfortunate misfit between signal lengths and favored FFT lengths, the number of arithmetic operations in the time-domain implementation is far less than those required by the frequency domain version: 514 versus $62,271$. If the input signal had been one sample shorter, the frequency-domain computations would have been more than a factor of two less $(28,696)$, but far more than in the time-domain implementation.

An interesting signal processing aspect of this example is demonstrated at the beginning and end of the output. The ramping up and down that occurs can be traced to assuming the input is zero before it begins and after it ends. The filter "sees" these initial and final values as the difference equation passes over the input. These artifacts can be handled in two ways: we can just ignore the edge effects or the data from previous and succeeding years' last and first week, respectively, can be placed at the ends.

To determine for what signal and filter durations a time- or frequency-domain implementation would be the most efficient, we need only count the computations required by each. For the time-domain, difference-equation approach, we need $N_x(2q+1)$. The frequency-domain approach requires three Fourier transforms, each requiring $\frac{K}{2}\log K$ computations for a length-K FFT, and the multiplication of two spectra ($6K$ computations). The output-signal-duration-determined length

---

The `filter` program has the "feature" that the length of its output equals the length of its input. To force it to produce a signal having the proper length, the program zero-pads the input appropriately.

must be at least $N_x + q$. Thus, we must compare

$$N_x(2q+1) \longleftrightarrow 6(N_x+q) + \frac{3}{2}(N_x+q)\log_2(N_x+q) \ .$$

Exact analytic evaluation of this comparison is quite difficult (we have a transcendental equation to solve). Insight into this comparison is best obtained by dividing by N_x.

$$(2q+1) \longleftrightarrow 6\left(1+\frac{q}{N_x}\right) + \frac{3}{2}\left(1+\frac{q}{N_x}\right)\log_2(N_x+q)$$

With this manipulation, we are evaluating the number of computations per sample. For any given value of the filter's order q, the right side, the number of frequency-domain computations, will exceed the left if the signal's duration is long enough. However, for filter durations greater than about 10, as long as the input is at least 10 samples, the frequency-domain approach is faster *so long as the FFT's power-of-two constraint is advantageous.*

The frequency-domain approach is not yet viable; what will we do when the input signal is infinitely long? The difference equation scenario fits perfectly with the envisioned digital filtering structure of Figure 5.12 $\{161\}$, but so far we have required the input to have limited duration (so that we could calculate its Fourier transform). The solution to this problem is quite simple: Section the input into frames, filter each, and add the results together. To section a signal means expressing it as a linear combination of length-N_x non-overlapping "chunks." Because the filter is linear, filtering a sum of terms is equivalent to summing the results of filtering each term.

$$x(n) = \sum_{m=-\infty}^{\infty} x(n-mN_x) \Longleftrightarrow y(n) = \sum_{m=-\infty}^{\infty} y(n-mN_x)$$

As illustrated in Fig. 5.16, note that each filtered section has a duration longer than the input. Consequently, we must literally add the filtered sections together, not just butt them together.

Computational considerations reveal a substantial advantage for a frequency-domain implementation over a time-domain one. The number of computations for a time-domain implementation essentially remains constant whether we section the input or not. Thus, the number of computations for each output is $2q+1$. In the frequency-domain approach, computation counting changes because we need only compute the filter's frequency response $H(k)$ once, which amounts to a fixed overhead. We need only compute two DFTs and multiply them to filter a section.

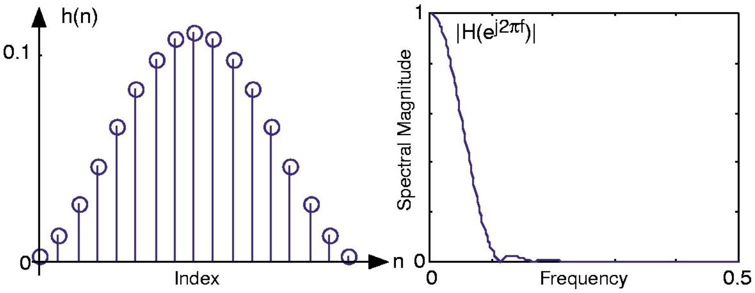


Figure 5.15 The figure shows the unit-sample response of a length-17 Hanning filter on the left and the frequency response on the right. This filter functions as a lowpass filter having a cutoff frequency of about 0.1.

Letting N_x denote a section's length, the number of computations for a section amounts to $(N_x+q)\log_2(N_x+q)+6(N_x+q)$. In addition, we must add the filtered outputs together; the number of terms to add corresponds to the excess duration of the output compared with the input (q). The frequency-domain approach thus requires $\left(1+\frac{q}{N_x}\right)\log_2(N_x+q)+7\frac{q}{N_x}+6$ computations per output value. For even modest filter orders, the frequency-domain approach is much faster.

Question 5.24 $\{257\}$ *Show that as the section length increases, the frequency domain approach becomes* increasingly*more efficient.*

Note that the choice of section duration is arbitrary. Once the filter is chosen, we should section so that the required FFT length is precisely a power of two: Choose N_x so that $N_x+q=2^\ell$.

Example

We want to lowpass filter a signal that contains a sinusoid and a significant amount of noise. Fig. 5.16 shows a portion of this signal's waveform. If it weren't for the overlaid sinusoid, discerning the sine wave in the signal is virtually impossible. One of the primary applications of linear filters is *noise removal*: preserve the signal by matching filter's passband with the signal's spectrum and greatly reduce all other frequency components that may be present in the noisy signal.

A smart Rice engineer has selected a FIR filter having a unit-sample response corresponding a period-17 sinusoid: $h(n)=\left(1-\cos 2\pi(n+1)/17\right)/18$, $n=0,\dots,16$, which makes $q=16$. Its frequency response (determined by computing the discrete Fourier transform) is shown in Fig. 5.15. To apply We can select the length of each section so that the frequency-domain filtering approach is maximally efficient: Choose the section length N_x so that N_x+q is a power of two. To use a length-64 FFT, each section must be 48 samples long. Filtering with the difference

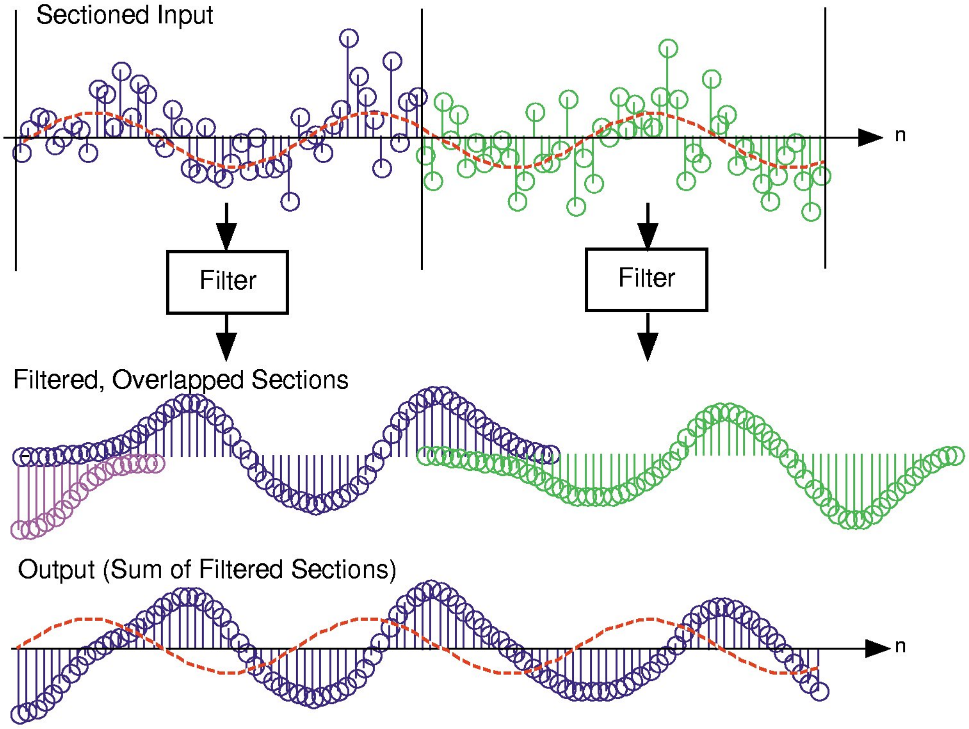


Figure 5.16 The noisy input signal is sectioned into length-48 frames, each of which is filtered using frequency-domain techniques. Each filtered section is added to other outputs that overlap to create the signal equivalent to having filtered the entire input. The sinusoidal component of the signal is shown as the red dashed line.

---

equation would require 33 computations per output while the frequency domain requires a little over 16; this frequency-domain implementation is over twice as fast! Fig. 5.16 shows how frequency-domain filtering works.

We note that the noise has been dramatically reduced, with a sinusoid now clearly visible in the filtered output. Some residual noise remains because noise components within the filter's passband appear in the output as well as the signal.

Question 5.25 {**257**} *Note that when compared to the input signal's sinusoidal component, the output's sinusoidal component seems to be delayed. What is the source of this delay? Can it be removed?*

Implementing the digital filter shown in Figure 5.12 {161} with a frequency-domain implementation requires some additional signal management not required by time-domain implementations. Conceptually, a real-time, time-domain filter could accept each sample as it becomes available, calculate the difference equation, and produce the output value, all in less that the sampling interval T_s. Frequency-domain approaches don't operate on a sample-by-sample basis; instead, they operate on sections. They filter in real time by producing N_x outputs for the same number of inputs faster than N_xT_s. Because they generally take longer to produce an output section than the sampling interval duration, we must filter one section while accepting into memory the *next* section to be filtered. In programming, the operation of building up sections while computing on previous ones is known as *buffering*. Buffering can also be used in time-domain filters as well but isn't required.

5.7 Modeling Real-World Systems with Artificial Ones

One aspect of discrete-time signal processing that has no counterpart in the analog world is developing models of systems to describe how signals arise. For example, the fundamental model of the physics of speech production amounts to passing a sequence of pulses through a linear system (Eq. (4.17) $\{116\}$). Meaning is conveyed by the properties of the linear system, which describes how the vocal tract responds to pitch pulses. Physically and mathematically, the speech production system is analog. However, suppose we sample the speech signal, obeying the Sampling Theorem of course. The discrete-time model for sampled speech would again be a sequence of pulses, this time a periodic sequence of unit samples, passing through a linear discrete-time system governed by some transfer function. It is this transfer function that we can *estimate* from the sampled speech signal, which means that we have the potential for extracting from the sampled speech signal what is being said. In short, we could develop a *speech recognition system*.

The key notion we need to develop is how to estimate a discrete-time system's input-output relation by observing its output with only partial knowledge of the input (in the speech case we know it's a sequence of pulses, but we don't know when they occur). Assume that the input-output relation we seek has the form of a difference equation with no "*b*" terms other than the first.[*]

$$s(n) = a_1 s(n-1) + \cdots + a_p s(n-p) + Gx(n)$$

What we develop in sequel is a way of estimating $a_1, \ldots, a_p$ and G from the observed signal $s(n)$.

---

[*] This case—having only one term in the difference equation involving the input —turns out to be the simplest. If you try to include more and follow the subsequent estimation procedure, you will discover that the situation is not nearly as nice.

Let's begin with the simplest case: $p = 1$. If the signal we indeed governed by the relation $s(n) = a_1 s(n-1) + Gx(n)$, with the input signal $x(n)$ having mostly zero values save when unit samples occur, then we should be able to *predict* each sample from the previous one. Letting $\widehat{s}(n)$ denote the predicted signal, it is given by $\widehat{s}(n) = a_1 s(n)$. However, we don't know the value of a_1; that's what we want to find by *minimizing the mean-squared prediction error*. Mathematically, we want to solve the problem

$$\min_{a_1} \sum_n [s(n) - \widehat{s}(n)]^2 \ .$$

Substituting the form of prediction made by our simple linear system, the problem becomes finding the value of a_1 to minimize $\sum_n [s(n) - a_1 s(n-1)]^2$. This minimization is easily accomplished by evaluating the derivative with respect to a_1, setting the result to zero, and solving for a_1.

$$\frac{d}{da_1} \sum_n [s(n) - a_1 s(n-1)]^2 = -2 \sum_n s(n-1)\, [s(n) - a_1 s(n-1)] = 0 \tag{5.14}$$

$$\Longrightarrow \boxed{a_1 = \frac{\sum_n s(n)s(n-1)}{\sum_n s^2(n)}} \tag{5.15}$$

Therefore, we can find the best model having the assumed form *from the signal itself*. For our simple case, the quantities that must be computed and the calculation of the model's parameters can easily be made.

More generally, we need to use more terms; we solve for the parameters $a_1, \ldots, a_p$ by minimizing the prediction error as before.

$$\min_{a_1,\ldots,a_p} \sum_n \left[s(n) - \sum_{m=1}^{p} a_m s(n-m) \right]^2$$

We must calculate the partial derivative with respect to each parameter, and set each derivative to zero, yielding a set of p equations to solve.

$$\frac{\partial}{\partial a_i} \sum_n \left[s(n) - \sum_{m=1}^{p} a_m s(n-m) \right]^2 = -2 \sum_n s(n-i) \left[s(n) - \sum_{m=1}^{p} a_m s(n-m) \right] , \ i = 1, \ldots, p$$

Simplifications lead to the following set of *linear* equations that must be solved to find $a_1, \ldots, a_p$.

$$\begin{array}{lllll}
a_1 \sum\limits_n s^2(n-1) & + a_2 \sum\limits_n s(n-1)s(n-2) & + \cdots + a_p \sum\limits_n s(n-1)s(n-p) & = \sum\limits_n s(n)s(n-1) \\
a_1 \sum\limits_n s(n-2)s(n-1) & + a_2 \sum\limits_n s^2(n-2) & + \cdots + a_p \sum\limits_n s(n-2)s(n-p) & = \sum\limits_n s(n)s(n-2) \\
\vdots & & & \vdots \\
a_1 \sum\limits_n s(n-p)s(n-1) & + a_2 \sum\limits_n s(n-p)s(n-2) & + \cdots + a_p \sum\limits_n s^2(n-p) & = \sum\limits_n s(n)s(n-p)
\end{array}$$

The coefficients in this equation set are known as *autocorrelation functions*, and are succinctly expressed as

$$r(i,j) = \sum_n s(n-i)s(n-j) \; .$$

To use matrix notation, let $\mathbf{R}$ denote the matrix formed from these autocorrelations ($\mathbf{R}_{i,j} = r(i,j)$), $\mathbf{a} = \mathrm{col}[a_1, \ldots, a_p]$, and $\mathbf{r} = \mathrm{col}[r(0,1), \ldots, r(0,p)]$. The equations that must be solved for our model's parameters now take the form

$$\mathbf{Ra} = \mathbf{r} \; .$$

We have p equations in p unknowns, and the solution is mathematically expressed using the matrix inverse.

$$\mathbf{a} = \mathbf{R}^{-1}\mathbf{r}$$

Analytically calculating this matrix inverse is quite difficult, In MATLAB, numerically solving such sets of linear equations is easy.

Once solved, the parameters thus obtained provide us with a system model, expressible with both the difference equation and the frequency response, for how the signal came to be. This technique has many important applications, all of which revolve around the fact that a few numbers, the filter parameters, summarize all the signal's essential features. Focusing on speech, we can use the parameters to provide a spectrum that reflects the vocal tract's characteristics; see Problem 5.15 $\{177\}$. We can use these parameters to concisely represent an entire speech section. Such a succinct, accurate representation means that we can efficiently represent essential speech features to communicate what is being said or to develop speech recognition systems.

Problems

5.1 Discrete-Time Fourier Transforms

Find the Fourier transforms of the following sequences, where $s(n)$ is some sequence having Fourier transform $S(e^{j2\pi f})$.

(a) $(-1)^n s(n)$

(b) $s(n) \cdot \cos 2\pi f_0 n$

(c) $x(n) = \begin{cases} s(n/2), & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$

(d) $n \cdot s(n)$

5.2 Discrete-Time Filtering

We can find the input-output relation for a discrete-time filter much more easily than for analog filters. The key idea is that a sequence can be written as a weighted linear combination of unit samples.

(a) Show that $x(n) = \sum_i x(i)\delta(n-i)$, where $\delta(n)$ is the unit-sample.

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise} \end{cases}$$

(b) If $h(n)$ denotes the *unit-sample response* —the output of a discrete-time linear, shift-invariant filter to a unit-sample input —find an expression for the output.

(c) In particular, assume our filter is FIR, with the unit-sample response having duration $q+1$. If the input has duration N, what is the duration of the filter's output to this signal?

(d) Let the filter be a boxcar averager: $h(n) = 1/(q+1)$, $n = 0, \ldots, q$, and zero otherwise. Let the input be a pulse of unit height and duration N. Find the filter's output when $N = (q+1)/2$.

5.3 A Special Discrete-Time Filter

Consider a FIR filter governed by the difference equation

$$y(n) = \frac{1}{3}x(n+2) + \frac{2}{3}x(n+1) + x(n) + \frac{2}{3}x(n-1) + \frac{1}{3}x(n-2)$$

(a) Find this filter's unit-sample response.

(b) Find this filter's transfer function. Characterize this transfer function (i.e., what classic filter category does it fall into).

(c) Suppose we take a sequence and stretch it out by a factor of three.

$$x(n) = \begin{cases} s(n/3), & n = 3m,\ m = \ldots, -1, 0, 1, \ldots \\ 0, & \text{otherwise} \end{cases}$$

Sketch the sequence $x(n)$ for some example $s(n)$. What is the filter's output to this input? In particular, what is the output at the indices where the input $x(n)$ is intentionally zero? Now how would you characterize this system?

5.4 The DFT

Let's explore the DFT and its properties.

(a) What is the length-K DFT of length-N boxcar sequence, where $N < K$?

(b) Consider the special case where $K = 4$. Find the inverse DFT of the product of the DFTs of two length-3 boxcars.

(c) If we could use DFTs to perform linear filtering, it should be true that the product of two DFTs equals the DFT of the filtered output. Imagine that one boxcar serves as the input and the other the coefficients of an FIR filter having a boxcar unit-sample response. The result of part (b) would then be the filter's output *if* we could implement the filter with length-3 DFTS. Does the acutal filtered output equal the result found in part (b)?

(d) What would you need to change so that the product of the DFTs of the input and unit-sample response in this case equaled the DFT of the filtered output?

5.5 The Fast Fourier Transform

Just to determine how fast the FFT algorithm really is, we can take advantage of MATLAB's `fft` function. If `x` is a length-N vector, `fft(x)` computes the length-N transform using the most efficient algorithm it can. In other words, it does not automatically zero-pad the sequence and it will use the FFT algorithm if the length is a power of two. Let's count the number of arithmetic operations the `fft` program requires for lengths ranging from 2 to 1024.

(a) For each length to be tested, generate a vector of random numbers, calculate the vector's transform, and count the number of floating-point operations (known as flops) it took. The following program illustrates the computations.

```
for n=2:1024
    x = randn(1,n);
    flops_start=flops;
    fft(x);
    count(n) = flops - flops_start;
end
```

(b) Plot the vector of computation counts. What lengths consume the most computations? What complexity do they seem to have? What lengths have the fewest computations?

5.6 Discrete Cosine Transform (DCT)

The discrete cosine transform of a length-N sequence is defined to be

$$S_c(k) = \sum_{n=0}^{N-1} s(n) \cos \frac{2\pi nk}{2N} \quad k = 0, \ldots, 2N-1 .$$

Note that the number of frequency terms is $2N - 1$.

(a) Find the inverse DCT.

(b) Does a Parseval's Theorem hold for the DCT?

(c) You choose to transmit information about the signal $s(n)$ according to the DCT coefficients. You could only send one, which one would you send?

5.7 Digital Filters

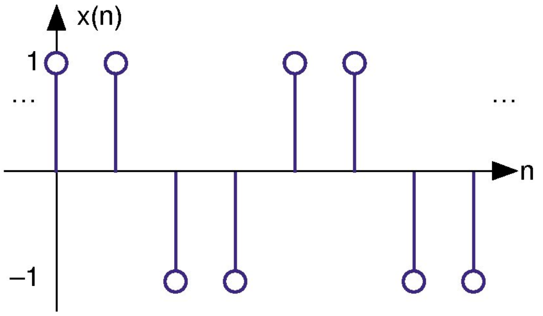
A filter has an input-output relationship given by the difference equation

$$y(n) = \frac{1}{4}x(n) + \frac{1}{2}x(n-1) + \frac{1}{4}x(n-2) \ .$$

(a) What is the filter's transfer function? How would you characterize it?

(b) What is the filter's output when the input equals $\cos \frac{\pi n}{2}$?

(c) What is the filter's output when the input is the depicted discrete-time square-wave?



5.8 Digital Filters

A discrete-time system is governed by the difference equation

$$y(n) = y(n-1) + \frac{x(n) + x(n-1)}{2} \ .$$

(a) Find the transfer function for this system.

(b) What is this system's output when the input is $\sin \pi n/2$?

(c) What is the input when the output is observed to be $y(n) = \delta(n) + \delta(n-1)$?

5.9 Detective Work

The signal $x(n)$ equals $\delta(n) - \delta(n-1)$.

(a) Find the length-8 DFT (discrete Fourier transform) of this signal.

(b) You are told that when $x(n)$ served as the input to a linear FIR (finite impulse response) filter, the output was

$$y(n) = \delta(n) - \delta(n-1) + 2\delta(n-2) \ .$$

Is this statement true? If so, indicate why and find the system's unit-sample response; if not, show why not.

5.10 A discrete-time, shift invariant, linear system produces an output $y(n) = 1, -1, 0, 0, \ldots$ when its input $x(n)$ equals a unit sample.

(a) Find the difference equation governing the system.

(b) Find the output when $x(n) = \cos 2\pi f_0 n$.

(c) How would you describe this system's function?

5.11 A discrete-time system has transfer function $H(e^{j2\pi f})$. A signal $x(n)$ is passed through this system to yield the signal $w(n)$. The time-reversed signal $w(-n)$ is then passed through the system to yield the time-reversed output $y(-n)$. What is the transfer function between $x(n)$ and $y(n)$?

5.12 **Removing "Hum"**
The slang word "hum" represents power line waveforms that creep into signals because of poor circuit construction. Usually, the 60 Hz signal (and its harmonics) are added to the desired signal. What we seek are filters that can remove hum. In this problem, the signal and the accompanying hum have been sampled; we want to design a *digital* filter for hum removal.

(a) Find filter coefficients for the length-3 FIR filter that can remove a sinusoid having *digital* frequency f_0 from its input.

(b) Assuming the sampling rate is f_s, to what analog frequency does f_0 correspond?

(c) A more general approach is to design a filter having a frequency response *magnitude* proportional to the absolute value of a cosine: $|H(e^{j2\pi f})| \propto |\cos \pi f N|$. In this way, not only can the fundamental but also its first few harmonics be removed. Select the parameter N and the sampling rate so that the frequencies at which the cosine equals zero correspond to 60 Hz and its odd harmonics through the fifth.

(d) Find the difference equation that defines this filter.

5.13 **Stock Market Data Processing**
Because a trading week lasts five days, stock markets frequently compute running averages each day over the previous five trading days to smooth price fluctuations. The technical stock analyst at the Buy-Lo–Sell-Hi brokerage firm has heard that FFT filtering techniques work better than any others (in terms of producing more accurate averages).

(a) What is the difference equation governing the five-day averager for daily stock prices?

(b) Design an efficient FFT-based filtering algorithm for the broker. How much data should be processed at once to produce an efficient algorithm? What length transform should be used?

(c) Is the analyst's information correct that FFT techniques produce more accurate averages than any others? Why or why not?

5.14 **Digital Filtering of Analog Signals**
RU Electronics wants to develop a filter that would be used in analog applications, but that is implemented digitally. The filter is to operate on signals that have a 10 kHz bandwidth, and will serve as a lowpass filter.

(a) What is the block diagram for your filter implementation? Explicitly denote which components are analog, which are digital (a computer performs the task), and which interface between analog and digital worlds.

(b) What sampling rate must be used and how many bits must be used in the A/D converter for the acquired signal's signal-to-noise ratio to be at least 60 dB? For this calculation, assume the signal is a sinusoid.

(c) If the filter is a length-128 FIR filter (the duration of the filter's unit-sample response equals 128), should it be implemented in the time or frequency domain?

(d) Assuming $H(e^{j2\pi f})$ is the transfer function of the digital filter, what is the transfer function of your system?

5.15 Linear Predictive Spectra

Among many interpretations of linear system modeling is its use to model how the signal could be produced. Suppose the sequence $y(n)$ is in fact governed by the difference equation

$$y(n) = a_1 x(n-1) + \cdots + a_p x(n-p) \ .$$

In this formulation, we must assume that the spectrum of the input $x(n)$ is roughly constant. We can use the linear predictive approach to estimate the filter's coefficients $a_1, \ldots, a_p$ and compute related quantities, such as spectra, as well.

(a) In terms of the filter coefficients, what is the model spectrum of $y(n)$?

(b) Load into Matlab the signal found in the file `y.mat` using the command `load /home/elec241/y.mat -ascii`. Use the MATLAB function `lpc` to calculate the linear predictive parameters for model order p of 6, 12, and 18: `a = lpc(y,p)`. Overlay the spectra inferred by the linear predictive model on the actual spectrum of $y(n)$. Contrast the two spectra for each model order. What features of the speech spectrum stand out better in the linear predictive spectrum than in the original? What is the effect of model order on the spectra? Estimate the locations of the first four formants from the linear predictive spectrum ($p = 18$).

Chapter 6

Information Communication

As far as a communications engineer is concerned, signals express information. Because systems manipulate signals, they also affect the information content. Information comes neatly packaged in both analog and digital forms. Speech, for example, is clearly an analog signal, and computer files consist of a sequence of bytes, a form of "discrete-time" signal despite the fact that the index sequences byte position, not time sample. *Communication systems* endeavor not to manipulate information, but to transmit it from one place to another, so-called *point-to-point* communication, from one place to many others, *broadcast* communication, or from many to many, like a telephone conference call or a chat room. Communication systems can be fundamentally analog, like radio, or digital, like computer networks.

This chapter develops a common theory that underlies how such systems work. We describe and analyze several such systems, some old like AM radio, some new like computer networks. The question as to which is better, analog or digital communication, has been answered, because of Claude Shannon's fundamental work on a theory of information published in 1948, the development of cheap, high-performance computers, and the creation of high-bandwidth communication systems. *The answer is to use a digital communication strategy.* In most cases, you should convert all information-bearing signals into discrete-time, amplitude-quantized signals. Fundamentally digital signals, like computer files (which are a special case of symbolic signals), are in the proper form. Because of the Sampling Theorem, we know how to convert analog signals into digital ones. Shannon showed that once in this form, *a properly engineered system can communicate digital information with no error despite the fact that the communication channel thrusts noise onto all transmissions.* This startling result has no counterpart in analog systems; AM radio will remain noisy. The convergence of these theoretical and engineering results on communications systems has had important consequences in other arenas. The audio compact disc (CD) and the digital videodisk (DVD) are now considered digital communications systems, with communication design considerations used throughout.

Go back to the fundamental model of communications, Figure 1.1 {4}. Communications design begins with two fundamental considerations.

- What is the nature of the information source, and to what extent can the receiver tolerate errors in the received information?
- What are the channel's characteristics and how do they affect the transmitted signal?

In short, what are we going to send and how are we going to send it? Interestingly, digital as well as analog transmission are accomplished using analog signals, like voltages in Ethernet (an example of *wireline* communications) and electromagnetic radiation (*wireless*) in cellular telephone. We begin with the channel . . .

6.1 Communication Channels

Electrical communications channels are either *wireline* or *wireless* channels (see §3.9 {53}). Wireline channels physically connect transmitter to receiver connection with a "wire," which could be twisted pair, coaxial cable or optic fiber. Consequently, wireline channels are more private and much less prone to interference. Simple wireline channels connect a single transmitter to a single receiver: a *point-to-point* connection as with the telephone. Some wireline channels operate in *broadcast* modes: one or more transmitter is connected to several receivers. One simple example of this situation is cable television. Computer networks can be found that operate in point-to-point or in broadcast modes. Wireless channels are much more public, with a transmitter's antenna radiating a signal that can be received by any antenna sufficiently close enough. In contrast to wireline channels where the receiver takes in only the transmitter's signal, the receiver's antenna will react to electromagnetic radiation coming from any source. This feature has two faces: The smiley face says that a receiver can take in transmissions from any source, letting receiver electronics select wanted signals and disregarding others, thereby allowing portable transmission and reception,* while the frowny face says that interference and noise are much more prevalent than in wireline situations. A noisier channel subject to interference compromises the flexibility of wireless communication.

---

*You will hear the term "tetherless networking" applied to completely wireless computer networks.

Long-distance transmission over either kind of channel encounters attenuation problems. We have explored losses in wireline channels in §3.9 {53}, where repeaters can extend the distance between transmitter and receiver beyond what passive losses the wireline channel imposes. In wireless channels, not only doe radiation losses occur ((3.5a) {61}), but also one antenna may not "see" another because of the earth's curvature. At the usual radio frequencies, propagating electromagnetic energy does not follow the earth's surface. *Line-of-sight* communication has the transmitter and receiver antennas in visual contact with each other. Assuming both antennas have height h above the earth's surface, maximum line-of-sight distance is

$$d_{\mathrm{LOS}} = 2\sqrt{2hR + h^2} \approx 2\sqrt{2Rh}\ ,$$

where R is the earth's radius (6.38×10^6 m).

Question 6.1 {**258**} *Derive the expression of line-of-sight distance using only the Pythagorean Theorem. Generalize it to the case where the antennas have different heights (as is the case with commercial radio and cellular telephone). What is the range of cellular telephone where the handset antenna has essentially zero height.*

Question 6.2 {**258**} *Can you imagine a situation wherein global wireless communication is possible with only one transmitting antenna? In particular, what happens to wavelength when carrier frequency decreases?*

Using a 100 m antenna would provide line-of-sight transmission over a distance of 71.4 km. Using such very tall antennas would provide wireless communication within a town or between closely spaced population centers. Consequently, *networks* of antennas sprinkle the countryside (each located on the highest hill possible) to provide long-distance wireless communications: Each antenna receives energy from one antenna and retransmits to another. This kind of network is known as a *relay network.*

If we were limited to line-of-sight communications, long distance wireless communication, like ship-to-shore communication, would be impossible. At the turn of the century, Marconi, the inventor of wireless telegraphy, boldly tried such long distance communication without any evidence —either empirical or theoretical — that it was possible. When the experiment worked, but only at night, physicists scrambled to determine why (using Maxwell's equations, of course). It was Oliver Heaviside, a mathematical physicist with strong engineering interests, who hypothesized that an invisible electromagnetic "mirror" surrounded the earth. What he meant was that at optical frequencies (and others as it turned out), the mirror was transparent, but at the frequencies Marconi used, it reflected electromagnetic radiation back to earth. He had predicted the existence of the ionosphere, a plasma that encompasses the earth at altitudes h_i between 80 and 180 km that reacts to solar radiation: It becomes transparent at Marconi's frequencies during the day, but becomes a mirror at night when solar radiation diminishes. The maximum distance along the earth's surface that can be reached by a *single* ionospheric reflection is $2R\cos^{-1}\left(R/(R+h_i)\right)$, which ranges between 2,010 and 3,000 km when we substitute minimum and maximum ionospheric altitudes. This distance does not span the United States or cross the Atlantic; for transatlantic communication, at least two reflections would be required. The communication delay encountered with a single reflection in this channel is $2\sqrt{2Rh_i + h_i^2}/c$, which ranges between 6.8 and 10 ms, again a small time interval.

Global wireless communication relies on satellites. Here, ground stations transmit to orbiting satellites that amplify the signal and retransmit it back to earth. Satellites will move across the sky unless they are in *geosynchronous orbits*, where the time for one revolution about the equator exactly matches the earth's rotation time of one day. TV satellites would require the homeowner to continually adjust his or her antenna if the satellite weren't in geosynchronous orbit. Newton's equations applied to orbiting bodies predict that the time T for one orbit is related to distance from the earth's center R as

$$R = \sqrt[3]{\frac{GMT^2}{4\pi^2}}\,.$$

where G is the gravitational constant and M the earth's mass. Calculations yield $R = 42,200$ km, which corresponds to an altitude of $35,700$ km. This altitude greatly exceeds that of the ionosphere, requiring satellite transmitters to use frequencies that pass through it. Of great importance in satellite communications is the transmission delay. The time for electromagnetic fields to propagate to a geosynchronous satellite and return is 0.24 s, a significant delay.

Question 6.3 {258} *In addition to delay, the propagation attenuation encountered in satellite communication far exceeds what occurs in ionospheric-mirror based communication. Calculate the attenuation incurred by radiation going to the satellite (one-way loss) with that encountered by Marconi (total going up and down). Note that the attenuation calculation in the ionospheric case, assuming the ionosphere acts like a perfect mirror, is* not *a straightforward application of the propagation loss formula {61}.*

6.1.1 Noise and Interference

We have mentioned that communications are, to varying degrees, subject to interference and noise. It's time to be more precise about what these quantities are and how they differ.

Interference represents man-made signals. Telephone lines are subject to power-line interference (in the United States a distorted 60 Hz sinusoid). Cellular telephone channels are subject to adjacent-cell phone conversations using the same signal frequency. The problem with such interference is that it occupies the same frequency band as the desired communication signal, and has a similar structure.

Question 6.4 **{259}** *Suppose interference occupied a different frequency band; how would the receiver remove it?*

We use the notation $i(t)$ to represent interference. Because interference has man-made structure, we can write an explicit expression for it that may contain some unknown aspects (how large it is, for example).

Noise signals have little structure and arise from both human and natural sources. Satellite channels are subject to deep space noise arising from electromagnetic radiation pervasive in the galaxy. Thermal noise plagues *all* electronic circuits that contain resistors. Thus, in receiving small amplitude signals, receiver amplifiers will most certainly add noise as they boost the signal's amplitude. All channels are subject to noise, and we need a way of describing such signals despite the fact we can't write a formula for the noise signal like we can for interference. The most widely used noise model is *white noise*. It is defined entirely by its frequency-domain characteristics.

- White noise has constant power at all frequencies.
- At each frequency, the phase of the noise spectrum is totally uncertain: It can be any value in between 0 and 2π, and its value at any frequency is unrelated to the phase at any other frequency.
- When noise signals arising from two different sources add, the resultant noise signal has a power equal to the sum of the component powers.

Because of the emphasis here on frequency-domain power, we are lead to define the *power spectrum*. Because of Parseval's Theorem (Eq. (4.14) {105}), we define the power spectrum $\mathcal{P}_s(f)$ of a non-noise signal $s(t)$ to be the magnitude-squared of its Fourier transform.

$$\mathcal{P}_s(f) \equiv |S(f)|^2 \tag{6.1}$$

Integrating the power spectrum over any range of frequencies equals the power the signal contains in that band. Because signals *must* have negative frequency components that mirror positive frequency ones, we routinely calculate the power in a spectral band as the integral over positive frequencies multiplied by two.

$$\text{Power in } [f_1, f_2] = 2\int_{f_1}^{f_2} \mathcal{P}_s(f)\, df$$

Using the notation $n(t)$ to represent a noise signal's waveform, we define noise in terms of its power spectrum. For white noise, the power spectrum equals the constant $\mathcal{N}_0/2$. With this definition, the power in a frequency band equals $\mathcal{N}_0(f_2 - f_1)$.

When we pass a signal through a linear, time-invariant system, the output's spectrum equals the product of the system's frequency response and the input's spectrum $\{109\}$. Thus, the power spectrum of the system's output is given by

$$\mathcal{P}_y(f) = |H(f)|^2 \mathcal{P}_x(f)\ .$$

This result applies to noise signals as well. When we pass white noise through a filter, the output is also a noise signal but with power spectrum $|H(f)|^2\mathcal{N}_0/2$.

6.1.2 Channel Models

Both wireline and wireless channels share characteristics, allowing us to use a common model for how the channel affects transmitted signals.

- The transmitted signal is usually not filtered by the channel.
- The signal can be attenuated.
- The signal propagates through the channel at a speed equal to or less than the speed of light, which means that the channel delays the transmission.
- The channel may introduce additive interference and/or noise.

Letting α represent the attenuation introduced by the channel, the receiver's input signal is related to the transmitted one by

$$\boxed{r(t) = \alpha x(t - \tau) + i(t) + n(t)} \tag{6.2}$$

This expression corresponds to the system model for the channel shown in Fig. 6.1. In this book, we shall assume that the noise is white.

Question 6.5 $\{$259$\}$ *Is this model for the channel linear?*

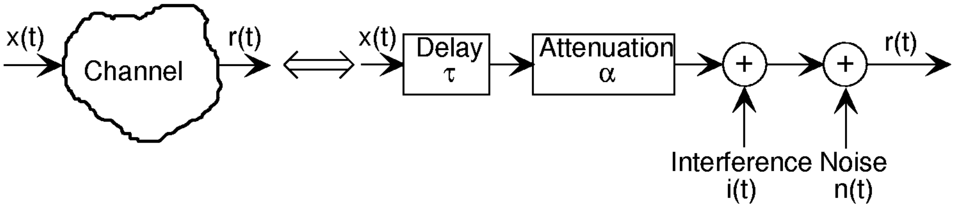


Figure 6.1 The channel component of the fundamental model of communication shown in Figure 1.1 {4} has the depicted form. The attenuation is due to propagation loss. Adding the interference and noise is justified by the linearity property of Maxwell's equations.

As expected, the signal that emerges from the channel is corrupted, but does contain the transmitted signal. Communication system design begins with detailing the channel model, then developing the transmitter and receiver that best compensate for the channel's corrupting behavior. We characterize the channel's quality by the signal-to-interference ratio (*SIR*) and the signal-to-noise ratio (*SNR*). The ratios are computed according to the relative power of each *within the transmitted signal's bandwidth.* Assuming the signal $x(t)$'s spectrum spans the frequency interval $[f_l, f_u]$, these ratios can be expressed in terms of power spectra.

$$SIR = \frac{\alpha^2 \int_0^{\infty} \mathcal{P}_x(f)\, df}{\int_{f_l}^{f_u} \mathcal{P}_i(f)\, df}$$

$$SNR = \frac{2\alpha^2 \int_0^{\infty} \mathcal{P}_x(f)\, df}{\mathcal{N}_0 \cdot (f_u - f_l)}$$

In most cases, the interference and noise powers do not vary for a given receiver. Variations in signal-to-interference and signal-to-noise ratios arise from the attenuation because of transmitter-to-receiver distance variations.

6.2 Analog Communication

We use analog communication techniques for analog message signals, like music, speech, and television. Transmission and reception of analog signals using analog results in an inherently noisy received signal (assuming the channel adds noise, which it almost certainly does).

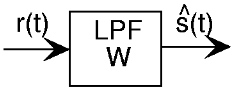


Figure 6.2 The receiver for baseband communication systems is quite simple: a lowpass filter having the same bandwidth as the signal.

6.2.1 Baseband communication

The simplest form of analog communication is *baseband communication.* Here, the transmitted signal equals the message times a transmitter gain.

$$x(t) = Gs(t)$$

An example, which is somewhat out of date, is the wireline telephone system. You don't use baseband communication in wireless systems simply because low-frequency signals do not radiate well. The receiver in a baseband system can't do much more than filter the received signal to remove out-of-band noise (interference is small in wireline channels). Assuming the signal occupies a bandwidth of W Hz (the signal's spectrum extends from zero to W), the receiver applies a lowpass filter having the same bandwidth (Fig. 6.2).

We use the *signal-to-noise ratio* of the receiver's output $\hat{s}(t)$ to evaluate any analog-message communication system. Assume that the channel introduces an attenuation α and white noise of spectral height $\mathcal{N}_0/2$. The filter does not affect the signal component —we assume its gain is unity —but does filter the noise, removing frequency components above W Hz. In the filter's output, the received signal power equals $\alpha^2 G^2 \, \mathrm{power}(s)$ and the noise power $\mathcal{N}_0 W$, which gives a signal-to-noise ratio of

$$SNR_{\mathrm{baseband}} = \frac{\alpha^2 G^2 \, \mathrm{power}(s)}{\mathcal{N}_0 W}$$

The signal power $\mathrm{power}(s)$ will be proportional to the bandwidth W; thus, in baseband communication the signal-to-noise ratio varies only with transmitter gain and channel attenuation and noise level.

6.2.2 Modulated Communication

Especially for wireless channels, like commercial radio and television, but also for wireline systems like cable television, the message signal must be *modulated*: The transmitted signal's spectrum occurs at much higher frequencies than those occupied by the signal. The key idea of modulation is to affect the amplitude, frequency or phase of what is known as the *carrier* sinusoid. Frequency modulation (FM) and less frequently used phase modulation (PM) are not discussed here; we

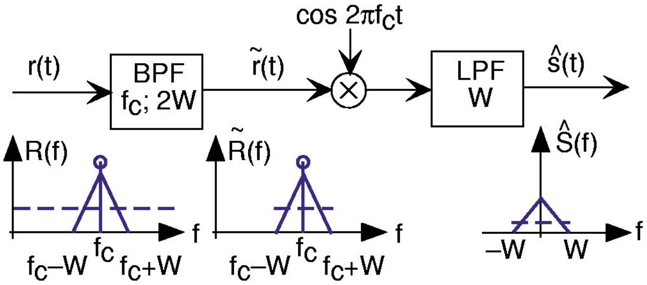


Figure 6.3 The AM coherent receiver along with the spectra of key signals is shown for the case of a triangular-shaped signal spectrum. The dashed line indicates the white noise level. Note that the filters' characteristics —cutoff frequency and center frequency for the bandpass filter—must be match to the modulation and message parameters.

focus on amplitude modulation (AM). The amplitude modulated message signal has the form

$$x(t) = A_c[1 + s(t)]\cos 2\pi f_c t \ ,$$

where f_c is the *carrier frequency* and A_c the carrier amplitude. Also, the signal's amplitude is assumed to be less than one: $|s(t)| < 1$. From our previous exposure to amplitude modulation (see the example $\{107\}$), we know that the transmitted signal's spectrum occupies the frequency range $[f_c - W, f_c + W]$, assuming the signal's bandwidth is W Hz (Figure 4.11 $\{108\}$). The carrier frequency is usually much larger than the signal's highest frequency —$f_c \gg W$, which means that the transmitter antenna and carrier frequency are chosen jointly during the design process.

Ignoring the attenuation and noise introduced by the channel for the moment, reception of an amplitude modulated signal is quite easy (see Problem 4.4 $\{125\}$). The so-called *coherent* receiver multiplies the input signal by a sinusoid and lowpass-filters the result (Fig. 6.3).

$$\widehat{s}(t) = \mathrm{LPF}\{x(t)\cdot\cos 2\pi f_c t\} = \mathrm{LPF}\{A_c[1 + s(t)]\cos^2 2\pi f_c t\}$$

Because of our trigonometric identities, we know that

$$\cos^2 2\pi f_c t = \frac{1}{2}(1 + \cos 2\pi 2 f_c t) \ .$$

At this point, the message signal is multiplied by a constant and a sinusoid at twice the carrier frequency. Multiplication by the constant term returns the message signal to baseband (where we want it to be!) while multiplication by the double-frequency term yields a very high frequency signal. The lowpass filter removes this high-frequency signal, leaving only the baseband signal. Thus, the received signal is

$$\widehat{s}(t) = \frac{A_c}{2}[1 + s(t)] \,.$$

Question 6.6 {**259**} *This derivation relies solely on the time domain; derive the same result in the frequency domain. You won't need the trigonometric identity with this approach.*

Because it is so easy to remove the constant term by electrical means —we insert a capacitor in series with receiver's output —we typically ignore it and concentrate on the signal portion of the receiver's output when calculating signal-to-noise ratio.

When we consider the much more realistic situation when we have a channel that introduces attenuation and noise, we can make use of the just-described receiver's linear nature to directly derive the receiver's output. The attenuation affects the output in the same way as the transmitted signal: It scales the output signal by the same amount. The white noise, on the other hand, should be filtered from the received signal before demodulation. We must thus insert a bandpass filter having bandwidth $2W$ and center frequency f_c: This filter has no effect on the received signal-related component, but does remove out-of-band noise power. As shown in Fig. 6.3, we apply coherent receiver to this filtered signal, with the result that the demodulated output contains noise that cannot be removed: It lies in the same spectral band as the signal.

As we derive the signal-to-noise ratio in the demodulated signal, let's also calculate the signal-to-noise ratio of the bandpass filter's output $\tilde{r}(t)$. The signal component of $\tilde{r}(t)$ equals $\alpha A_c s(t) \cos 2\pi f_c t$. This signal's Fourier transform equals

$$\frac{\alpha A_c}{2}\left(S(f+f_c) + S(f-f_c)\right) \,,$$

making the power spectrum

$$\frac{\alpha^2 A_c^2}{4}\left(|S(f+f_c)|^2 + |S(f-f_c)|^2\right)$$

Question 6.7 {**259**} *If you calculate the magnitude-squared of the first equation, you don't obtain the second* in the general case. *What is the previous assumption that makes the power spectrum calculation correct?*

Thus, the total signal-related power in $\tilde{r}(t)$ is $\frac{\alpha^2 A_c^2}{2}\,\mathrm{power}[s]$. The noise power equals the integral of the noise amplitude $\mathcal{N}_0$ times the filter's bandwidth $2W$.

The so-called *received signal-to-noise ratio* —the signal-to-noise ratio after the *de rigeur* front-end bandpass filter and before demodulation —equals

$$SNR_r = \frac{\alpha^2 A_c^2 \,\mathrm{power}[s]}{4\mathcal{N}_0 W}\ .$$

The demodulated signal $\widehat{s}(t) = \alpha A_c s(t)/2 + n_{\mathrm{out}}(t)$. Clearly, the signal power equals $\alpha^2 A_c^2\,\mathrm{power}[s]/4$. To determine the noise power, we must understand how the coherent demodulator affects the bandpass noise found in $\tilde{r}(t)$. Because we are concerned with noise, we must deal with the power spectrum since we don't have the Fourier transform available to us. Letting $\mathcal{P}_{\tilde{n}}(f)$ denote the power spectrum of $\tilde{r}(t)$'s noise component, the power spectrum after multiplication by the carrier has the form

$$\frac{\mathcal{P}_{\tilde{n}}(f+f_c)+\mathcal{P}_{\tilde{n}}(f-f_c)}{4}\ .$$

The delay and advance in frequency indicated here results in two spectral noise bands falling in the low-frequency region of lowpass filter's passband. Thus, the total noise power in this filter's output equals $2\cdot\mathcal{N}_0/2\cdot 2W\cdot 1/4 = \mathcal{N}_0 W/2$. The signal-to-noise ratio of the receiver's output thus equals

$$\boxed{SNR_{\widehat{s}} = \frac{\alpha^2 A_c^2\,\mathrm{power}[s]}{2\mathcal{N}_0 W} = 2SNR_r}\ .$$

Let's break down the components of this signal-to-noise ratio to better appreciate how the channel and the transmitter parameters affect communications performance. Better performance, as measured by the *SNR*, occurs as it increases.

- More transmitter power — increasing A_c — increases the signal-to-noise ratio proportionally.
- The carrier frequency f_c has no effect on *SNR*, but we have assumed that $f_c \gg W$.
- The signal bandwidth W enters the signal-to-noise expression in two places: implicitly through the signal power and explicitly in the expression's denominator. *If the signal spectrum had a constant amplitude* as we increased the bandwidth, signal power would increase proportionally. On the other hand, our transmitter enforced the criterion that signal *amplitude* was constant $\{187\}$. Signal amplitude essentially equals the integral of the magnitude of the signal's spectrum.* Enforcing the signal amplitude specification

---

*This result isn't exact, but we do know that $s(0) = \int_{-\infty}^{\infty} S(f)\,df$.

means that as the signal's bandwidth increases we must decrease the spectral amplitude, with the result that the signal power remains constant. Thus, increasing signal bandwidth does indeed decrease the signal-to-noise ratio of the receiver's output.

- Increasing channel attenuation—moving the receiver farther from the transmitter—decreases the signal-to-noise ratio as the square. Thus, signal-to-noise ratio decreases as distance-squared between transmitter and receiver.
- Noise added by the channel adversely affects the signal-to-noise ratio.

In summary, amplitude modulation provides an effective means for sending a bandlimited signal from one place to another. For wireline channels, using baseband or amplitude modulation makes little difference in terms of signal-to-noise ratio. For wireless channels, amplitude modulation is the only alternative. The one AM parameter that does not affect signal-to-noise ratio is the carrier frequency f_c: We can choose any value we want so long as the transmitter and receiver use the same value. However, suppose someone else wants to use AM and chooses the same carrier frequency. The two resulting transmissions will add, and *both* receivers will produce the sum of the two signals. What we clearly need to do is talk to the other party, and agree to use separate carrier frequencies. As more and more users wish to use radio, we need a forum for agreeing on carrier frequencies and on signal bandwidth.

Question 6.8 {**259**} *Suppose all users agree to use the same signal bandwidth. How closely can the carrier frequencies be while avoiding communications crosstalk? What is the signal bandwidth for commercial AM? How does this bandwidth compare to the speech bandwidth?*

On earth, this forum is the government. In the United States, the Federal Communications Commission (FCC) strictly controls the use of the electromagnetic spectrum for communications. Separate frequency bands are allocated for commercial AM, FM, cellular telephone (the analog version of which is AM), short wave (also AM), and satellite communications.

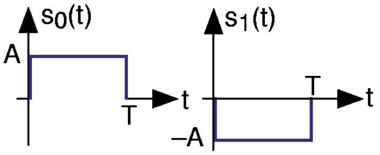
6.3 Digital Communication

Effective, error-free transmission of a sequence of bits —a *bit stream* —is the goal here. We found that analog schemes, as represented by amplitude modulation, always yield a received signal containing noise as well as the message signal when the channel adds noise. Digital communication schemes are very different. Once we decide how to represent bits by analog signals that can be transmitted over wireline (like a computer network) or wireless (like digital cellular telephone) channels, we will then develop a way of tacking on communication bits to the message bits that will reduce channel-induced errors greatly. In theory, digital communication errors can be zero, even though the channel adds noise!

6.3.1 Signal Sets

We represent a bit by associating one of two specific analog signals with the bit's value. Thus, if $b(n) = \mathtt{0}$, we transmit the signal $s_0(t)$; if $b(n) = \mathtt{1}$, send $s_1(t)$. These two signals comprise the *signal set* for digital communication and are designed with the channel and bit stream in mind. In virtually every case, these signals have a finite duration T common to both signals that is known as the *bit interval.* A commonly used example of a signal set consists of pulses that are negatives of each other.

$$\begin{aligned} s_0(t) &= A\, \mathrm{p}_T(t) \\ s_1(t) &= -A\, \mathrm{p}_T(t) \end{aligned}$$



Here, we have a baseband signal set suitable for wireline transmission. The entire bit stream $b(n)$ is represented by a sequence of these signals. Mathematically, the transmitted signal has the form

$$x(t) = \sum_n (-1)^{b(n)} A\, \mathrm{p}_T(t - nT) \tag{6.3}$$

and graphically Fig. 6.4 shows what a typical transmitted signal might be. This way of representing a bit stream —changing the bit changes the sign of the transmitted signal —is known as *binary phase shift keying* and abbreviated BPSK. Here is an attempt to explain the nomenclature: The word binary is clear enough (one binary-valued quantity is transmitted during a bit interval); changing the sign of sinusoid amounts to changing —shifting —the phase by π (although we don't have a sinusoid yet); and the word "keying" reflects back to the first electrical communication system, also for digital communications: the telegraph.

The *datarate* R of a digital communication system is how frequently an information bit is transmitted. In this example equals the reciprocal of the bit interval: $R = 1/T$. Thus, for a 1 Mbps (megabit per second) transmission, we must have $T = 1\ \mu$s.

The choice of signals to represent bit values is arbitrary to some degree. Clearly, we do not want to choose signal set members to be the same; we couldn't distin-

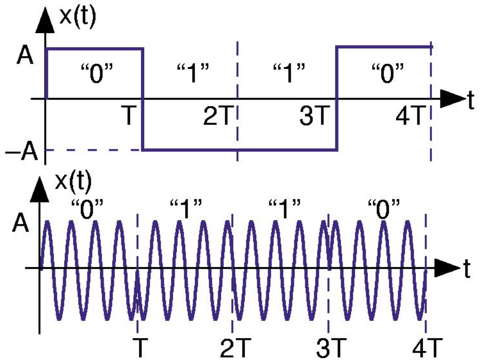
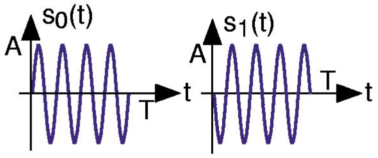


Figure 6.4 The upper plot shows how a baseband signal set for transmitting the bit sequence `0110`. The lower one shows an amplitude-modulated variant suitable for wireless channels.

guish bits if we did so. We could also have made the negative-amplitude pulse represent a `0` and the positive one a `1`. This choice is indeed arbitrary and will have not effect on performance *assuming* the receiver knows which signal represents which bit. As in all communication systems, we design transmitter and receiver together.

A simple signal set for a wireless channel amounts to amplitude modulating the baseband signal set just given by a carrier having a frequency harmonic with the bit interval.

$$
\begin{aligned}
s_0(t) &= A\,\mathrm{p}_T(t)\sin\left(\frac{2\pi kt}{T}\right)\\
s_1(t) &= -A\,\mathrm{p}_T(t)\sin\left(\frac{2\pi kt}{T}\right)
\end{aligned}
$$



Question 6.9 {259} *What is the value of k in this example?*

This signal set is also known as a BPSK signal set. We'll show later that indeed both signal sets provide identical performance levels when the signal-to-noise ratios are equal.

Question 6.10 {259} *Write a formula, in the style of* (6.3)*, for the transmitted signal as shown in Figure 6.4* {*192*} *that emerges when we use this modulated signal.*

What is the transmission bandwidth of these signal sets? We need only consider the baseband version as the second is an amplitude-modulated version of the first. The bandwidth is determined by the bit sequence. If the bit sequence is

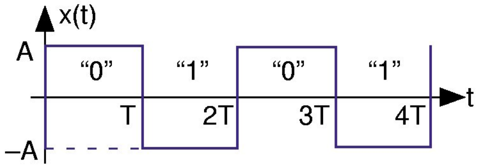


Figure 6.5 Here we show the transmitted waveform corresponding to an alternating bit sequence.

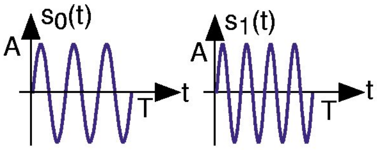
constant—always `0` or always `1`—the transmitted signal is a constant, which has zero bandwidth. The worst-case—bandwidth consuming—bit sequence is the alternating one shown in Fig. 6.5. In this case, the transmitted signal is a square wave having a period of $2T$. From our work in Fourier series, we know that this signal's spectrum contains odd-harmonics of the fundamental, which here equals $1/2T$. Thus, strictly speaking, the signal's bandwidth is infinite. In practical terms, we use the 90%-power bandwidth to assess the effective range of frequencies consumed by the signal. The first and third harmonics contain that fraction of the total power, meaning that the effective bandwidth of our baseband signal is $3/2T$ or, expressing this quantity in terms of the datarate, $3R/2$. Thus, a digital communications signal requires more bandwidth than the datarate: a 1 Mbps *baseband* system requires a bandwidth of at least 1.5 MHz. Listen carefully when someone describes the transmission bandwidth of digital communication systems: Did they say "megabits" or "megahertz"?

Question 6.11 {260} *Show that indeed the first and third harmonics contain 90% of the transmitted power. If the receiver uses a front-end filter of bandwidth* $3/2T$*, what is the total harmonic distortion of the received signal?*

Question 6.12 {260} *What is the 90% transmission bandwidth of the modulated signal set?*

A different signal set from the previous two uses the bit to affect the frequency of a carrier sinusoid.

$$s_0(t) = A\,\mathrm{p}_T(t)\sin 2\pi f_0 t$$
$$s_1(t) = A\,\mathrm{p}_T(t)\sin 2\pi f_1 t$$



where usually the frequencies f_0, f_1 are harmonically related to the bit interval. In the depicted example, $f_0 = 3/T$ and $f_1 = 4/T$. This signal set is known as *frequency shift keying* or FSK. As can be seen from the transmitted signal for our

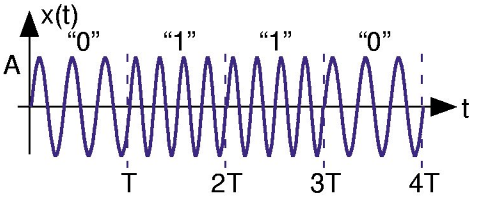


Figure 6.6 This plot shows the FSK waveform for same bitstream used in Figure 6.4 {192}.

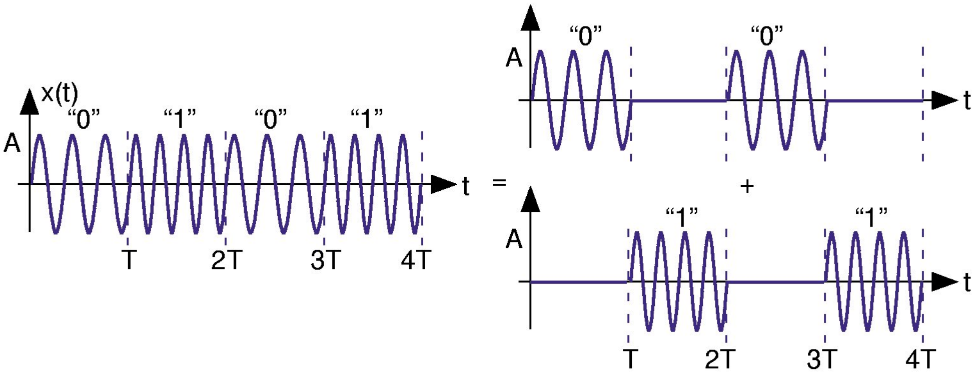


Figure 6.7 The depicted decomposition of the FSK-modulated alternating bit stream into its frequency components simplifies the calculation of its bandwidth.

example bit stream (Fig. 6.6), the transitions at bit interval boundaries are smoother than those of BPSK (Figure 6.4 {192}).

To determine the bandwidth required by this signal set, we again consider the alternating bit stream. Think of it as two signals added together: The first comprised of the signal $s_0(t)$, the zero signal, $s_0(t)$, zero, etc., and the second having the same structure but interleaved with the first (Fig. 6.7). Each component can be thought of as a fixed-frequency sinusoid multiplied by a square wave of period $1/2T$ that alternates between one and zero. This baseband square wave has the same Fourier spectrum as our BPSK example, but with the addition of the constant term c_0. This quantity's presence changes the number of Fourier series terms required for the 90% bandwidth: Now we need only include the zero and first harmonics to achieve it. The bandwidth thus equals (with $f_0 < f_1$) $f_1 + 1/2T - (f_0 - 1/2T) = f_1 - f_0 + 1/T$. This bandwidth is larger than the BPSK bandwidth.

Question 6.13 {260} *Show that the bandwidth for FSK is always larger. What is*

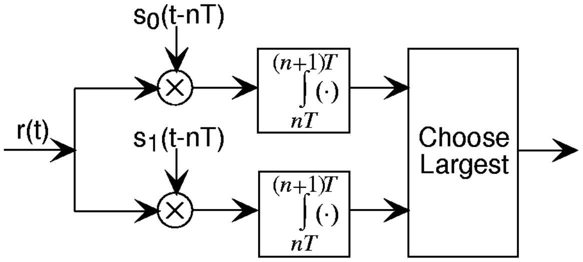


Figure 6.8 The *optimal* receiver structure for digital communication faced with additive white noise channels is the depicted matched filter.

the smallest it can be?

6.3.2 Digital Communication Receivers

The receiver interested in the transmitted bit stream must perform two tasks when received waveform $r(t)$ begins.

1. It must determine when bit boundaries occur: The receiver needs to *synchronize* with the transmitted signal. Because transmitter and receiver are designed in concert, both use the same value for the bit interval T. Synchronization can occur because the transmitter begins sending with a reference bit sequence, known as the *preamble*. The receiver knows the preamble. The receiver must use this reference sequence, which is usually the alternating sequence like shown in Figs. 6.5 {193} and 6.7 {194}, to determine when bit boundaries occur. This procedure amounts to what in digital hardware as *self-clocking signaling*: The receiver of a bit stream must derive the clock —when bit boundaries occur —from its input signal. The transmitter signals the end of the preamble by switching to a second bit sequence. Because the receiver usually does not determine which bit was sent until synchronization occurs, it does not know when during the preamble it obtained synchronization. The second preamble phase informs the receiver that data bits are about to come and that the preamble is almost over.
2. Once synchronized and data bits are transmitted, the receiver must then determine every T seconds what bit was transmitted during the previous bit interval. We focus on this aspect of the digital receiver because this strategy is also used in synchronization.

The receiver for digital communication is known as a *matched filter*. This receiver, shown in Fig. 6.8, multiplies the received signal by each of the possible members of the transmitter signal set, integrates the product over the bit interval, and compares the results. Whichever path through the receiver yielded the largest value corresponds to the receiver's decision as to what bit was sent during the previous bit interval. For the next bit interval, the multiplication and integration begins again, with the next bit decision made at the end of the bit interval. Mathematically, the received value of $b(n)$, which we label $\widehat{b}(n)$, is given by*

$$\boxed{\widehat{b}(n) = \arg\max_i \int_{nT}^{(n+1)T} r(t) s_i(t)\, dt} \; .$$

Note that the precise numerical value of the integrator's output does not matter; what does matter is its value relative to the other integrator's output.

Let's assume a perfect channel for the moment: The received signal equals the transmitted one. If bit `0` were sent using the baseband BPSK signal set, the integrator outputs would be

$$\begin{aligned} \int_{nT}^{(n+1)T} r(t) s_0(t)\, dt &= A^2 T \\ \int_{nT}^{(n+1)T} r(t) s_1(t)\, dt &= -A^2 T \end{aligned}$$

If bit `1` were sent,

$$\begin{aligned} \int_{nT}^{(n+1)T} r(t) s_0(t)\, dt &= -A^2 T \\ \int_{nT}^{(n+1)T} r(t) s_1(t)\, dt &= A^2 T \end{aligned}$$

Question 6.14 {**260**} *Can you develop a receiver for BPSK signal sets that requires only* one *multiplier-integrator combination?*

Question 6.15 {**260**} *What is the corresponding result when the amplitude-modulated BPSK signal set is used?*

Clearly, this receiver would always choose the bit correctly. Channel attenuation would not affect this correctness; it would only make the values smaller, but all that matters is which is largest.

---

*You may not have seen the $\arg\max_i$ notation before. $\max_i$ yields the maximum value of its argument with respect to the index i. $\arg\max_i$ equals the value of the index obtaining that maximum.

When we incorporate additive noise into our channel model, so that $r(t) = \alpha s_i(t) + n(t)$, errors can creep in. If the transmitter sent bit `0`, the integrators' outputs would be

$$\int_{nT}^{(n+1)T} r(t)s_0(t)\,dt = \alpha A^2 T + \int_{nT}^{(n+1)T} n(t)s_0(t)\,dt \qquad (6.4a)$$

$$\int_{nT}^{(n+1)T} r(t)s_1(t)\,dt = -\alpha A^2 T + \int_{nT}^{(n+1)T} n(t)s_1(t)\,dt \qquad (6.4b)$$

It is the quantities containing the noise terms that cause errors in the receiver's decision-making process. Because they involve noise, the values of these integrals are random quantities drawn from some probability distribution that vary erratically from bit interval to bit interval. Because the noise has zero average value and has an equal amount of power in all frequency bands, the values of the integrals will hover about zero. What is important is how much they vary. If the noise is such that its integral term is more negative than $\alpha A^2 T$, then the receiver will make an error, deciding that the transmitted zero-valued bit was indeed a one. The probability that this situation occurs depends on three factors:

- **Signal set choice**
  The *difference* between the signal-dependent terms in the integrators' outputs (6.4a), (6.4b) defines how large the noise term must be for an incorrect receiver decision to result. What affects the probability of such errors occurring is the square of this difference in comparison to the noise term's variability. For our BPSK baseband signal set, the signal-related value is $4\alpha^2 A^4 T^2$.
- **Variability of the noise term**
  We quantify variability by the average value of its square, which is essentially the noise term's power. This calculation is best performed in the frequency domain and equals

  $$\text{power}\left[\int_{nT}^{(n+1)T} n(t)s_0(t)\,dt\right] = \int_{-\infty}^{\infty} \frac{\mathcal{N}_0}{2}|S_0(f)|^2\,df$$

  Because of Parseval's Theorem, we know that $\int_{-\infty}^{\infty}|S_0(f)|^2\,df = \int_{-\infty}^{\infty} s_0^2(t)\,dt$, which for the baseband signal set equals $A^2 T$. Thus, the noise term's power is $\mathcal{N}_0 A^2 T/2$.
- **Probability distribution of the noise term**
  The value of the noise terms relative to the signal terms and the probability of their occurrence directly affect the likelihood that a receiver error will

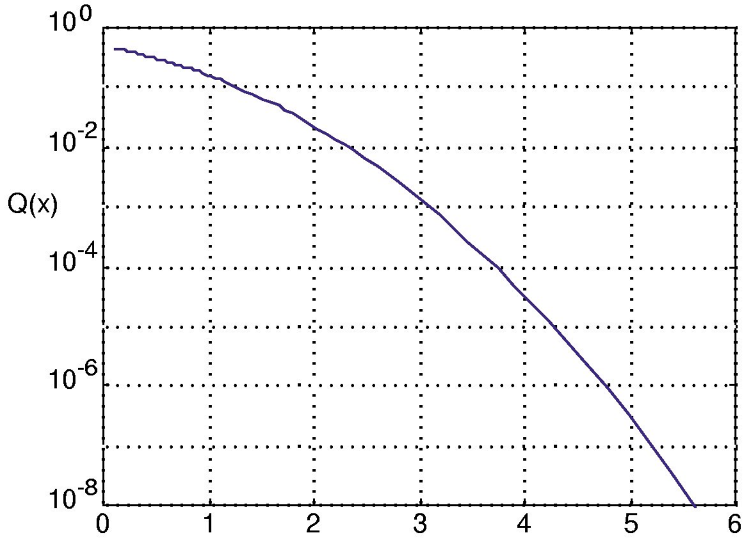


Figure 6.9 The function $Q(x)$ is plotted in semilogarithmic coordinates. Note that it decreases very rapidly for small increases in its arguments. For example, when x increases from 4 to 5, $Q(x)$ decreases by a factor of 100.

---

occur. For the white noise we have been considering, the underlying distributions are Gaussian. The probability the receiver makes an error on any bit transmission equals

$$p_e = Q\left(\frac{1}{2}\sqrt{\frac{\text{square of signal difference}}{\text{noise term power}}}\right) \tag{6.5}$$

$$p_e = Q\left(\sqrt{\frac{2\alpha^2 A^2 T}{\mathcal{N}_0}}\right)$$

where $Q(\cdot)$ is the integral

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\alpha^2/2}\, d\alpha \tag{6.6}$$

This integral has no closed form expression, but it can be accurately computed. As Fig. 6.9 illustrates, $Q(\cdot)$ is a decreasing, very nonlinear function.

The term $A^2 T$ equals the energy expended by the transmitter in sending the bit; we label this term E_b. We arrive at a concise expression for the probability the matched

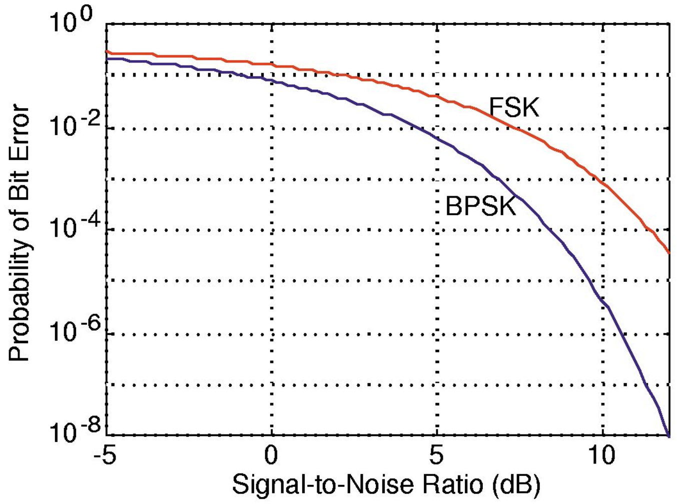


Figure 6.10 The probability that the matched-filter receiver makes an error on any bit transmission is plotted against the signal-to-noise ratio of the received signal. The upper curve shows the performance of the FSK signal set, the lower (and therefore better) one the BPSK signal set.

filter receiver makes a bit-reception error.

$$\boxed{p_e = Q\left(\sqrt{\frac{2\alpha^2 E_b}{\mathcal{N}_0}}\right)} \tag{6.7}$$

Fig. 6.10 shows how the receiver's error rate varies with the signal-to-noise ratio $\alpha^2 E_b/\mathcal{N}_0$.

Question 6.16 {**260**} *Derive the probability of error expression for the modulated BPSK signal set, and show that its performance identically equals that of the baseband BPSK signal set.*

This result reveals several properties about digital communication systems.

- As the received signal becomes increasingly noisy, whether due to increased distance from the transmitter (smaller α) or to increased noise in the channel (larger $\mathcal{N}_0$), the probability the receiver makes an error approaches $1/2$. In such situations, the receiver performs only slightly better than the "receiver" that ignores what was transmitted and merely guesses what bit was transmitted. Consequently, it becomes almost impossible to communicate information when digital channels become noisy.

- As the signal-to-noise ratio increases, performance gains —smaller probability of error p_e —can be easily obtained. At a signal-to-noise ratio of 12 dB, the probability the receiver makes an error equals 10^{-8}. In words, one out of one hundred million bits will, on the average, be in error.
- Once the signal-to-noise ratio exceeds about 5 dB, the error probability decreases dramatically. Adding 1 dB improvement in signal-to-noise ratio can result in a factor of 10 smaller p_e.
- Signal set choice can make a significant difference in performance. All BPSK signal sets, baseband or modulated, yield the same performance for the same bit energy. The BPSK signal set does perform much better than the FSK signal set once the signal-to-noise ratio exceeds about 5 dB.

Question 6.17 $\{$**260**$\}$ *Derive the expression for the probability of error that would result if the FSK signal set were used.*

The matched-filter receiver provides impressive performance once adequate signal-to-noise ratios occur. You might wonder whether another receiver might be better. The answer is that the matched-filter receiver is optimal: *No other receiver can provide a smaller probability of error than the matched filter regardless of the SNR.* Furthermore, no signal set can provide better performance than the BPSK signal set, where the signal representing a bit is the negative of the signal representing the other bit. The reason for this result rests in the dependence of probability of error p_e on the difference between the noise-free integrator outputs: For a given E_b, no other signal set provides a greater difference.

How small should the error probability be? Out of N transmitted bits, on the average Np_e bits will be received in error. Do note the phrase "on the average" here: Errors occur randomly because of the noise introduced by the channel, and we can only predict the probability of occurrence. Since bits are transmitted at a rate R, errors occur at an average frequency of Rp_e. Suppose the error probability is an impressively small number like 10^{-6}. Data on a computer network like Ethernet is transmitted at a rate $R = 10$ Mbps, which means that errors would occur roughly 100 per second. This error rate is very high, requiring a much smaller p_e to achieve a more acceptable average occurrence rate for errors occurring. Because Ethernet is a wireline channel, which means the channel noise is small and the attenuation low, obtaining very small error probabilities is not difficult. We do have some tricks up our sleeves, however, that can essentially reduce the error rate to zero *without* resorting to expending a large amount of energy at the transmitter. We need to understand *digital channels* and Shannon's Noisy Channel Coding theorem.

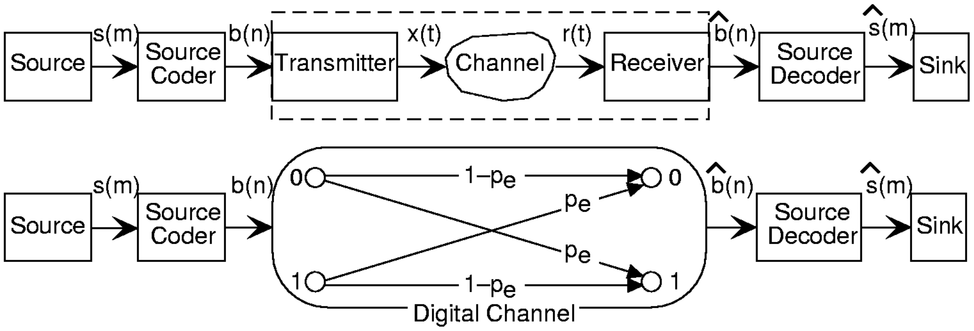


Figure 6.11 The steps in transmitting digital information are shown in the upper system, the Fundamental Model of Communication. The symbolic-valued signal $s(m)$ forms the message, and it is encoded into a bit sequence $b(n)$. The indices differ because more than one bit/symbol is usually required to represent the message by a bitstream. Each bit is represented by an analog signal, transmitted through the (unfriendly) channel, and received by a matched-filter receiver. From the received bitstream $\hat{b}(n)$ the received symbolic-valued signal $\hat{s}(m)$ is derived. The lower block diagram shows an equivalent system wherein the analog portions are combined and modeled by a transition diagram, which shows how each transmitted bit could be received. For example, transmitting a `0` results in the reception of a `1` with probability p_e (an error) or a `0` with probability $1 - p_e$ (no error).

6.3.3 Digital Channels

Let's review how digital communication systems work within the Fundamental Model of Communication. As shown in Fig. 6.11, the message is a single bit. The entire analog transmission/reception system, which we have determined in the previous section, can be lumped into a single system known as the digital channel. Digital channels are described by *transition diagrams*, which indicate the output alphabet symbols that result for each possible transmitted symbol and the probabilities of the various reception possibilities. The probabilities on transitions coming from the same symbol must sum to one. For the matched-filter receiver and the signal sets we have seen, the depicted transition diagram, known as a *binary symmetric channel*, captures how transmitted bits are received. The probability of error p_e is the sole parameter of the digital channel, and it encapsulates signal set choice, channel properties, and the matched-filter receiver. With this simple but entirely accurate model, we can concentrate on how bits are received.

6.3.4 Digital Sources and Compression

Communication theory has been formulated best for symbolic-valued signals. Claude Shannon published in 1948 *The Mathematical Theory of Communication*, which became the cornerstone of digital communication. He showed the power of *probabilistic models* for symbolic-valued signals, which allowed him to quantify the information present in a signal. In the simplest signal model, each symbol can occur at index n with a probability $\Pr[a_k]$, $k = 0, \dots, K-1$. What this model says is that for each signal value a K-sided coin is flipped (note that the coin need not be fair). For this model to make sense, the probabilities must be numbers between zero and one and must sum to one.

$$0 \le \Pr[a_k] \le 1$$

$$\sum_{k=1}^{K} \Pr[a_k] = 1$$

This coin-flipping model assumes that symbols occur without regard to what preceding or succeeding symbols were, a false assumption for typed text. Despite this probabilistic model's over-simplicity, the ideas we develop here also work when more accurate, but still probabilistic, models are used. The key quantity that characterizes a symbolic-valued signal is the *entropy* of its alphabet.

$$\boxed{\mathcal{H}[\mathcal{A}] = -\sum_{k} \Pr[a_k] \log_2 \Pr[a_k]} \tag{6.8}$$

Because we use the base-2 logarithm, entropy has units of bits. For this definition to make sense, we must take special note of symbols having probability zero of occurring. A zero-probability symbol never occurs; thus, we define $0 \log_2 0 = 0$ so that such symbols do not affect the entropy. The maximum value attainable by an alphabet's entropy occurs when the symbols are equally likely ($\Pr[a_k] = \Pr[a_l]$, $k \neq l$). In this case, the entropy equals $\log_2 K$. The minimum value occurs when only one symbol occurs; it has probability one of occurring and the rest have probability zero.

Question 6.18 {261} *Derive the maximum-entropy results, both the numeric aspect (entropy equals* $\log_2 K$*) and the theoretical one (equally likely symbols maximize entropy). Derive the value of the* minimum-*entropy alphabet.*

Example

A four-symbol alphabet has the following probabilities.

$$\Pr[a_0] = \frac{1}{2} \quad \Pr[a_1] = \frac{1}{4} \quad \Pr[a_2] = \frac{1}{8} \quad \Pr[a_3] = \frac{1}{8}$$

Note that these probabilities sum to one as they should. As $\frac{1}{2} = 2^{-1}$, $\log_2 \frac{1}{2} = -1$. The entropy of this alphabet equals

$$\begin{aligned}\mathcal{H}[\mathcal{A}] &= -(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{8}\log_2\frac{1}{8} + \frac{1}{8}\log_2\frac{1}{8})\\ &= -(\frac{1}{2}(-1) + \frac{1}{4}(-2) + \frac{1}{8}(-3) + \frac{1}{8}(-3))\\ &= 1.75 \text{ bits}\end{aligned}$$

The significance of an alphabet's entropy rests in how we can represent it with a sequence of *bits*. Bit sequences form the "coin of the realm" in digital communications: They are the universal way of representing symbolic-valued signals. We convert back and forth between symbols to bit-sequences with what is known as a *codebook*:* A table that associates symbols to bit sequences. In creating this table, we must be able to assign a *unique* bit sequence to each symbol so that we can go between symbol and bit sequence without error.

As we shall explore in some detail later (§6.3 {190}), digital communication is the transmission of symbolic-valued signals from one place to another. When faced with the problem, for example, of sending a file across the Internet, we must first represent each character by a bit sequence. Because we want to send the file quickly, we want to use as few bits as possible. However, we don't want to use so few bits that the receiver cannot determine what each character was from the bit sequence. For example, we could use one bit for every character: File transmission would be fast but useless because the codebook creates errors. Shannon proved in his monumental work what we call today the *Source Coding Theorem*. Let $B(a_k)$ denote the number of bits used to represent the symbol a_k. The average number of bits $\overline{B}(\mathcal{A})$ required to represent the entire alphabet equals $\sum_k B(a_k) \Pr[a_k]$. *The Source Coding Theorem states that the average number of bits needed to represent* accurately *the alphabet need only to satisfy*

$$\mathcal{H}[\mathcal{A}] \leq \overline{B}(\mathcal{A}) < \mathcal{H}[\mathcal{A}] + 1 \ . \tag{6.9}$$

Thus, the alphabet's entropy specifies to within one bit how many bits on the average need to be used to send the alphabet. The smaller an alphabet's entropy, the fewer bits required for digital transmission of files expressed in that alphabet.

---

[*] You may be conjuring the notion of hiding information from others when we use the name codebook for the symbol-to-bit-sequence table. There is no relation to cryptology, which comprises mathematically provable methods of securing information. The codebook terminology was developed during the beginnings of information theory just after World War II.

Example

Continuing our example, let's see if we can find a codebook for our four-letter alphabet that satisfies the Source Coding Theorem. The simplest code to try is known as the simple binary code: Convert the symbol's index into a binary number and use the same number of bits for each symbol by including leading zeros where necessary.

$$a_0 \leftrightarrow 00 \qquad a_1 \leftrightarrow 01 \qquad a_2 \leftrightarrow 10 \qquad a_3 \leftrightarrow 11$$

Whenever the number of symbols in the alphabet is a power of two (as in this case), the average number of bits $\overline{B}(\mathcal{A})$ equals $\log_2 K$, which equals 2 in this case. Because the entropy equals 1.75 bits, the simple binary code indeed satisfies the Source Coding Theorem—we are within one bit of the entropy limit—but you might wonder if you can do better. If we chose a codebook with differing number of bits for the symbols, a smaller average number of bits can indeed be obtained. The idea is to use shorter bit sequences for the symbols that occur more often. One codebook like this is

$$a_0 \leftrightarrow 0 \qquad a_1 \leftrightarrow 10 \qquad a_2 \leftrightarrow 110 \qquad a_3 \leftrightarrow 111$$

Now $\overline{B}(\mathcal{A}) = 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = 1.75$ bits. We can reach the entropy limit! The simple binary code is, in this case, less efficient than the unequal-length code. Using the efficient code, we can transmit the symbolic-valued signal having this alphabet 12.5% faster. Furthermore, we know that no more efficient codebook can be found because of Shannon's Theorem.

Another interpretation of Shannon's result rests in *data compression*. Here, we have a symbolic-valued signal source, like a computer file or an image, that we want to represent with as few bits as possible. Compression schemes that assign symbols to bit sequences are known as *lossless* if they obey the Source Coding Theorem; they are *lossy* if they use fewer bits than the alphabet's entropy. Using a lossy compression scheme means that you cannot recover a symbolic-valued signal from its compressed version without incurring some error. You might be wondering why anyone would want to intentionally create errors, but lossy compression schemes are frequently used where the efficiency gained in representing the signal outweighs the significance of the errors.

Shannon's Source Coding Theorem (Eq. (6.9) $\{203\}$) states that symbolic-valued signals require *on the average* at least $\mathcal{H}[\mathcal{A}]$ number of bits to represent each its values, which are symbols drawn from the alphabet $\mathcal{A}$. We also found that using a so-called *fixed rate* source coder, one that produces a fixed number

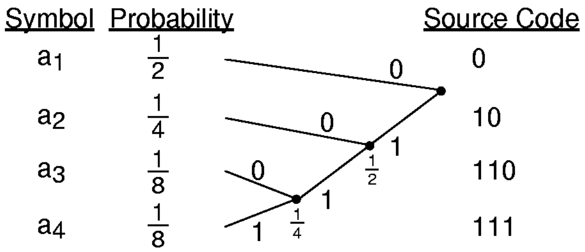


Figure 6.12 We form a Huffman code for a four-letter alphabet having the indicated probabilities of occurrence. The binary tree created by the algorithm extends to the right, with the root node (the one at which the tree begins) defining the codewords. The bit sequence obtained by traversing the tree from the root to the symbol defines that symbol's binary code.

of bits/symbol, may not be the most efficient way of encoding symbols into bits. What was not discussed there was a procedure for designing an efficient source coder: one *guaranteed* to produce the fewest bits/symbol on the average. That source coder is not unique, and one approach that does achieve that limit is the *Huffman source coding algorithm.*[*]

1. Create a vertical table for the symbols, the best ordering being in decreasing order of probability.
2. Form a binary tree to the right of the table. A binary tree always has two branches at each node. Build the tree by merging the two lowest probability symbols at each level, making the probability of the node equal to the sum of the merged nodes' probabilities. If more than two nodes/symbols share the lowest probability at a given level, pick any two; your choice won't affect $\overline{B}(\mathcal{A})$.
3. At each node, label each of the emanating branches with a binary number. The bit sequence obtained from passing from the tree's root to the symbol is its Huffman code.

Example

Our simple four-symbol alphabet used in previous examples {202},{204} has the Huffman coding tree shown in Fig. 6.12. The code thus obtained is not unique as we could have labeled the branches coming out of each node differently. The average number of bits required to represent this alphabet equals 1.75 bits, which is the Shannon entropy limit for this source alphabet. If we had the symbolic-valued signal $s(m) = \{a_2, a_3, a_1, a_4, a_1, a_2, \dots\}$, our Huffman code would produce the bitstream $b(n) = \texttt{101100111010}\cdots$.

---

[*] In the early years of information theory, the race was on to be the first to find a *provably* maximally efficient source coding algorithm. The race was won by then MIT graduate student David Huffman in 1954, who worked on the problem as a project in his information theory course. We're pretty sure he received an "A."

If the alphabet probabilities were different, clearly a different tree, and therefore different code, could well result. Furthermore, we may not be able to achieve the entropy limit. If our symbols had the probabilities $\Pr[a_1] = 1/2$, $\Pr[a_2] = 1/4$, $\Pr[a_3] = 1/5$, and $\Pr[a_4] = 1/20$, the average number of bits/symbol resulting from the Huffman coding algorithm equals 1.75. However, the entropy limit is 1.68 bits. The Huffman code does satisfy the Source Coding Theorem —its average length is within one bit of the alphabet's entropy —but you might wonder if a better code existed. David Huffman showed mathematically that no other code could achieve a shorter average code than his. We can't do better.

Question 6.19 $\{$**261**$\}$ *Derive the Huffman code for this second set of probabilities, and verify the claimed average code length and alphabet entropy.*

Because the bit sequences that represent individual symbols can have differing lengths, the bitstream index m does not increase in lock step with the symbol-valued signal's index n. To capture how often bits must be transmitted to keep up with the source's production of symbols, we can only compute averages. If our source code averages $\overline{B}(\mathcal{A})$ bits/symbol and symbols are produced at a rate R, the average bit rate equals $\overline{B}(\mathcal{A})R$, and this quantity determines the bit interval duration T.

Question 6.20 $\{$**261**$\}$ *Calculate what the relation between T and the average bit rate $\overline{B}(\mathcal{A})R$.*

A subtly of source coding is the whether we need "commas" in the bitstream. When we use an unequal number of bits to represent symbols, how does the receiver determine when symbols begin and end? If you created a source code that *required* a separation marker in the bitstream between symbols, it would be very inefficient since you are essentially requiring an extra symbol in the transmission stream.* Note that in the example, no commas where placed in the bitstream, but you can unambiguously decode the sequence of symbols from the bitstream. Huffman showed that his (maximally efficient) code had the *prefix* property: No code for a symbol began another symbol's code. Once you have the prefix property, the bitstream is *partially* self-synchronizing: Once the receiver knows where the bitstream starts, we can assign a unique and correct symbol sequence to the bitstream.

---

*A good example of this need is the Morse Code: Between each letter, the telegrapher needs to insert a pause to inform the receiver when letter boundaries occur.

Question 6.21 {**261**} *Sketch an argument that prefix coding, whether derived from a Huffman code or not, will provide unique decoding when an unequal number of bits/symbol are used in the code.*

However, having a prefix code does not guarantee total synchronization: After hopping into the middle of a bitstream, can we always find the correct symbol boundaries? The self-synchronization issue does mitigate the use of efficient source coding algorithms.

Question 6.22 {**261**} *Show by example that a bitstream produced by a Huffman code is not necessarily self-synchronizing. Are fixed-length codes self synchronizing?*

Another issue is bit errors induced by the digital channel; if they occur (and they will), synchronization can easily be lost even if the receiver started "in synch" with the source. Despite the small probabilities of error proffered by good signal set design and the matched filter, an infrequent error can devastate the ability to translate a bitstream into a symbolic signal. We need ways of reducing reception errors *without* demanding that p_e be smaller.

Example

The first electrical communications system —the telegraph —was digital. When first deployed in 1844, it communicated text over wireline connections using a binary code—the Morse code—to represent individual letters. To send a message from one place to another, telegraph operators would tap the message using a telegraph key to another operator, who would relay the message on to the next operator, presumably getting the message closer to its destination. In short, the telegraph relied on a *network* not unlike the basics of modern computer networks. To say it presaged modern communications would be an understatement. It was also far ahead of some needed technologies, namely the Source Coding Theorem. The Morse Code, shown in prettyreftablep:morse, was not a prefix code. To separate codes for each letter, Morse code required that a space—a pause—be inserted between each letter. In information theory, that space counts as another code letter, which means that the Morse code encoded text with a three-letter source code: dots, dashes and space. The resulting source code is not within a bit of entropy, and is grossly inefficient (about 25%). The table shows a Huffman code for english text, and as we know it *is* efficient.

	%	*Morse Code*	*Huffman Code*		%	*Morse Code*	*Huffman Code*
A	6.22	.-	1011	N	5.73	-.	0100
B	1.32	-...	010100	O	6.06	—	1000
C	3.11	-.-.	10101	P	1.87	.–.	00000
D	2.97	-..	01011	Q	0.10	–.-	0101011100
E	10.53	.	001	R	5.87	.-.	0111
F	1.68	..-.	110001	S	5.81	...	0110
G	1.65	–.	110000	T	7.68	-	1101
H	3.63	....	11001	U	2.27	..-	00010
I	6.14	..	1001	V	0.70	...-	0101010
J	0.06	.—	01010111011	W	1.13	.–	000011
K	0.31	-.-	01010110	X	0.25	-..-	010101111
L	3.07	.-..	10100	Y	1.07	-.–	000010
M	2.48	–	00011	Z	0.06	–..	0101011101011

Table 6.1 Morse and Huffman Codes for American-Roman Alphabet. The % column indicates the average probability (expressed in percent) of the letter occurring in english. The entropy $\mathcal{H}[\mathcal{A}]$ of the this source is 4.14 bits. The average Morse codeword length is 2.51 symbols. Adding one more symbol for the letter separator and converting to bits yields an average codeword length of 5.56 bits. The average Huffman codeword length is 4.35 bits.

6.3.5 Error-Correcting Codes: Channel Coding

We can, to some extent, *correct* errors made by the receiver with only the error-filled bit stream emerging from the digital channel available to us. The idea is for the transmitter to send not only the symbol-derived bits emerging from source coder but also additional bits derived from the coder's bit stream. These additional bits help the receiver determine if an error has occurred in the data bits (the important bits) or in the error-correction bits. Instead of the communication model shown in Figure 6.11 {201}, the transmitter inserts a *channel coder* before analog modulation, and the receiver the corresponding channel decoder (Fig. 6.13). *This block diagram shown there forms the Fundamental Model of Digital Communication.*

Perhaps the simplest error correcting code is the *repetition code*. Here, the transmitter sends the data bit several times, an odd number of times in fact. Because the error probability p_e is always less than $1/2$, we know that more of the bits should be correct rather than in error. Simple majority voting by the bits (hence the reason for the odd number) determines what the transmitted bit more accurately than sending it alone. For example, let's consider the three-fold repetition code.

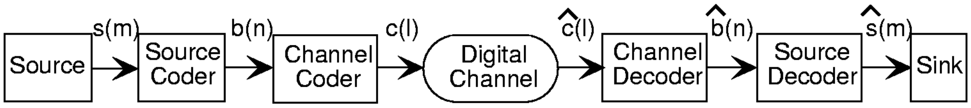


Figure 6.13 To correct errors that occur in the digital channel, a channel coder and decoder are added to the communication system. Properly designed channel coding can greatly reduce the probability (from the uncoded value of p_e) that a data bit $b(n)$ is received incorrectly even when the probability of $c(l)$ be received in error remains p_e or becomes larger. This system forms the Fundamental Model of *Digital* Communication.

Thus, for every bit $b(n)$ emerging from the source coder, the channel coder produces three. Thus, the bit stream emerging from the channel coder $c(l)$ has a data rate three times higher than that of the original bit stream $b(n)$. Letting p_e be the average bit error rate induced by the digital channel, the following table illustrates when errors can be corrected and when they can't. We use the transmission of $b(n) = 0 \rightarrow c(3n) = c(3n+1) = c(3n+2) = 0$ in this example.

$\hat{c}(l)$	*Probability*	$\hat{b}(n)$
000	$(1-p_e)^3$	0
001	$p_e(1-p_e)^2$	0
010	$p_e(1-p_e)^2$	0
011	$p_e^2(1-p_e)$	1
100	$p_e(1-p_e)^2$	0
101	$p_e^2(1-p_e)$	1
110	$p_e^2(1-p_e)$	1
111	p_e^3	1

Thus, if one bit of the three bits is received in error, the receiver can correct the error; if more than one error occurs, the channel decoder announces the bit is 1 instead of transmitted value of 0. Using this repetition code, the probability of $\hat{b}(n) \neq 0$ equals $3p_e^2(1-p_e) + p_e^3$. This probability of a bit error is always less than p_e, the uncoded value, so long as $p_e < 1/2$.

Question 6.23 {261} *Demonstrate mathematically that this claim is indeed true.*

The repetition code represents a special case of what is known as *block channel coding*. For every K bits that enter the block channel coder, it inserts an additional $N-K$ error-correction bits to produce a block of N bits for transmission. We use the notation (N, K) to represent a given block code's parameters. In our three-fold repetition code, $K=1$ and $N=3$. A block code's *coding efficiency* E equals the ratio K/N, and quantifies the overhead introduced by channel coding. The rate at which bits must be transmitted again changes: So-called data bits $b(n)$ emerge from the source coder at an average rate $\overline{B}(\mathcal{A})R$ and exit the channel at a rate $1/E$ higher. We represent the fact that the bits sent through the digital channel operate at a different rate by using the index l for the channel-coded bit stream $c(l)$. Note that the blocking (framing) imposed by the channel coder does not correspond to symbol boundaries in the bit stream $b(n)$, especially when we employ variable-length source codes.

Because of the higher datarate imposed by the channel coder, the probability of bit error occurring in the digital channel *increases* relative to the value obtained when no channel coding is used. The bit interval duration must be reduced by K/N in comparison to the no-channel-coding situation, which means the energy per bit E_b goes *down* by the same amount.[*] Because of this reduction, the error probability p_e of the digital channel goes up. The question thus becomes does channel coding *really* help: Is the effective error probability lower with channel coding even though the error probability for each transmitted bit is larger? The answer is *no*: Using a repetition code for channel coding cannot ultimately reduce the probability that a data bit is received in error. The ultimate reason is the repetition code's inefficiency: Its efficiency E of $1/3$ is too small for the amount of error correction it provides.

Question 6.24 {262} *Using* MATLAB, *calculate the probability a bit is received incorrectly with a three-fold repetition code. Show that when the energy per bit E_b is reduced by $1/3$ that this probability is larger than the no-coding probability of error.*

Does any error-correcting code reduce communication errors when real-world constraints are taken into account? The answer now is yes. To understand channel coding, we need to develop first a general framework for channel coding, and discover what it takes for a code to be maximally efficient: Correct as many errors as possible using the fewest error correction bits as possible (making the efficiency K/N as large as possible).

So-called *linear* codes create error-correction bits by combining the data bits linearly. The phrase "linear combination" means here single-bit binary arithmetic.

$$\begin{array}{llll} 0 \oplus 0 = 0 & 0 \oplus 1 = 1 & 0 \cdot 0 = 0 & 0 \cdot 1 = 0 \\ 1 \oplus 1 = 0 & 1 \oplus 0 = 1 & 1 \cdot 1 = 1 & 1 \cdot 0 = 0 \end{array} \tag{6.10}$$

For example, let's consider the specific $(7,4)$ error correction code described by the following coding table and, more concisely, by the succeeding matrix expression.

---

[*] It is unlikely that the transmitter's power could be increased to compensate. Such is the sometimes-unfriendly nature of the real world.

$$\begin{aligned}
c(1) &= b(1)\\
c(2) &= b(2)\\
c(3) &= b(3)\\
c(4) &= b(4)\\
c(5) &= b(1) \oplus b(2) \oplus b(3)\\
c(6) &= b(2) \oplus b(3) \oplus b(4)\\
c(7) &= b(1) \oplus b(2) \oplus b(4)
\end{aligned}$$

or

$$\mathbf{c} = \mathbf{G}\mathbf{b}, \quad \mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ \cdots & \cdots & \cdots & \cdots\\ 1 & 1 & 1 & 0\\ 0 & 1 & 1 & 1\\ 1 & 1 & 0 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} c(1)\\ c(2)\\ c(3)\\ c(4)\\ c(5)\\ c(6)\\ c(7) \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b(1)\\ b(2)\\ b(3)\\ b(4) \end{bmatrix} \qquad (6.11)$$

The length-K block of data bits is represented by the vector $\mathbf{b}$, and the length-N output block of the channel coder, known as a *codeword*, by $\mathbf{c}$. For example, if $\mathbf{b} = \text{col}[1, 0, 0, 0]$, $\mathbf{c} = \text{col}[1, 0, 0, 0, 1, 0, 1]$. The *generator matrix* $\mathbf{G}$ defines all block-oriented linear channel coders.

In this $(7, 4)$ code, $2^4 = 16$ of the $2^7 = 128$ possible blocks arriving at the channel decoder correspond to error-free transmission and reception. In analyzing the error-correcting capability of this or any other channel code, we must have a systematic way of assigning a codeword to a received block when communication errors occur. Note that the number of possible received blocks is much larger than the number of codewords; the channel has ample opportunity to confuse us! Conceptually, we want to assign to each possible received length-N block a length-K block of data bits. This assignment should correspond to the most probable error pattern that could have transformed a transmitted block into the one received. Because the matched-filter receiver guarantees that the single-bit error probability p_e of the digital channel will be less than $1/2$, the most probable error pattern is the one having the *fewest* errors. The way of assessing error patterns is to compute the *distance* between the received block and all codewords. The *Hamming distance* between blocks $\mathbf{c}_1$ and $\mathbf{c}_2$, denoted by $d(\mathbf{c}_1, \mathbf{c}_2)$, equals the sum of the difference between the bit sequences comprising the two blocks. For example, if $\mathbf{c}_1 = \text{col}[\texttt{1},\texttt{0},\texttt{0},\texttt{0},\texttt{1},\texttt{0},\texttt{1}]$ and $\mathbf{c}_2 = \text{col}[\texttt{1},\texttt{0},\texttt{1},\texttt{0},\texttt{1},\texttt{0},\texttt{1}]$, the difference equals $\text{col}[\texttt{0},\texttt{0},\texttt{1},\texttt{0},\texttt{0},\texttt{0},\texttt{0}]$, the number of nonzero terms equals one, and thus $d(\mathbf{c}_1, \mathbf{c}_2) = 1$. Interestingly, examination of the binary arithmetic table $\{210\}$ reveals that subtraction and addition are equivalent. We can express the Hamming distance as

$$d(\mathbf{c}_1, \mathbf{c}_2) = \text{sum}(\mathbf{c}_1 \oplus \mathbf{c}_2) \ .$$

Error correction amounts to searching for the codeword $\mathbf{c}$ closest to the received block $\widehat{\mathbf{c}}$ in terms of the Hamming distance between the two. The error correction capability of a channel code is limited by how close together any two error-free blocks are. Bad codes would produce blocks close together, which would result in ambiguity when assigning a block of data bits to a received block. The quantity to examine, therefore, in designing code error correction codes is the minimum distance between codewords.

$$d_{\text{min}} = \min_{\mathbf{c}_i \neq \mathbf{c}_j} d(\mathbf{c}_i, \mathbf{c}_j)$$

Suppose this minimum distance is one; this means the channel code has two codewords with the not-so-desirable property that one error occurring somewhere within the N-bit block transforms one codeword into another! The receiver would never realize that an error occurred in this case. Such a channel code would have no error-correction capability, and because the energy/bit must be reduced to insert error correction bits, we obtain larger error probabilities. Now suppose the minimum distance between error-free blocks is two. This situation is better than the previous one: If one error occurs anywhere within a block, no matter what codeword was transmitted, we can detect the fact that an error occurred. Unfortunately, we *cannot* correct the error. Because the minimum distance is two, a received block containing one error can be a distance of one from two (or more) codewords; we cannot associate such a block *uniquely* with a codeword. We must be resigned to the fact that to correct errors, we must have the minimum distance between error-free blocks be at least three.

Question 6.25 $\{262\}$ *Suppose we want a channel code to have an error-correction capability of n bits. What must the minimum Hamming distance between codewords d_{min} be?*

How do we calculate the minimum distance between codewords? Because we have 2^K codewords, the number of possible unique pairs equals $2^{K-1}(2^K - 1)$, which can be a large number. Recall that our channel coding procedure is linear, with $\mathbf{c} = \mathbf{G}\mathbf{b}$. Therefore $\mathbf{c}_i \oplus \mathbf{c}_j = \mathbf{G}(\mathbf{b}_i \oplus \mathbf{b}_j)$. Because $\mathbf{b}_i \oplus \mathbf{b}_j$ always yields another block of *data* bits, we find that the difference between any two codewords is another codeword! Thus, to find d_{min} we need only compute the number of ones that comprise all non-zero codewords. Finding these codewords is easy once we examine the coder's generator matrix. Note that the columns of $\mathbf{G}$ are codewords, and that all codewords can be found by all possible pairwise sums of the columns. To find d_{min}, we need only count the number of bits in each column and sums of columns. For our example $(7, 4)$ code (Eq. (6.11) $\{211\}$), $\mathbf{G}$'s first column has three ones, the next one four, and the last two three. No sum of columns has fewer than three bits, meaning $d_{\text{min}} = 3$, and we have a channel coder that can correct all occurrences of one error within a received 7-bit block.

Question 6.26 $\{$**262**$\}$ *Find the generator matrix for our three-fold repetition code. Show that it indeed has single-bit error correction capability. How many repetitions would be required to yield a double-error-correcting channel code?*

As with the repetition code, we must question whether our $(7, 4)$ code's error correction capability compensates for the increased error probability due to the necessitated reduced bit energy. Fig. 6.14 shows that if the signal-to-noise ratio is large enough that channel coding indeed yields a smaller overall error probability. Because the bit stream emerging from the source coder is segmented into four-bit blocks, the fair way of comparing coded and uncoded transmission is to compute the probability of a *block* error: the probability that any bit in a block remains in error despite error correction and regardless of whether the error occurs in the data or coding bits. Clearly, our $(7, 4)$ channel code does yield smaller error rates, and is worth the additional systems required to make it work.

Error-Correcting Codes: Channel Decoding

Because the idea of channel coding has merit (so long as the code is efficient), let's develop a systematic procedure for performing channel decoding. One way of checking for errors is to try recreating the error correction bits from the data portion of the received block $\widehat{\mathbf{c}}$. Using matrix notation, we make this calculation by multiplying the received block $\widehat{\mathbf{c}}$ by the matrix $\mathbf{H}$ known as the *parity check matrix*. It is formed from the generator matrix $\mathbf{G}$ by taking the bottom, error-correction

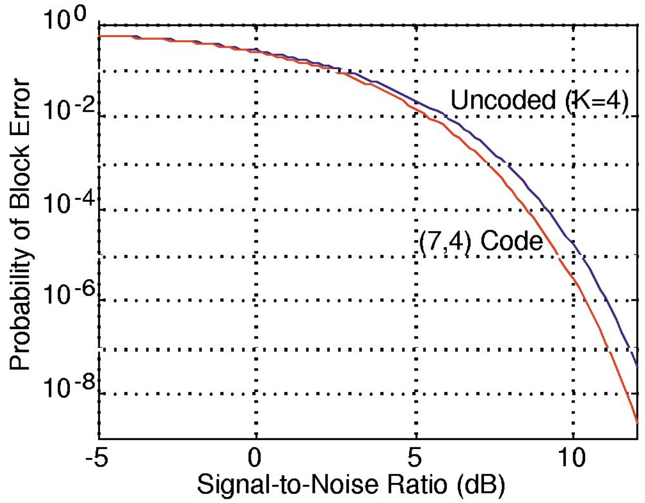


Figure 6.14 The probability of an error occurring in transmitted $K = 4$ data bits equals $1 - (1 - p_e)^4$ as $(1 - p_e)^4$ equals the probability that the four bits are received without error. The upper curve displays how this probability of an error anywhere in the four-bit block varies with signal-to-noise ratio. When a $(7, 4)$ single-bit error correcting code is used, the transmitter reduces the energy it expends during a single-bit transmission by $4/7$, appending three extra bits for error correction. Now the probability of any bit in the *seven*-bit block being in error after error correction equals $1 - (1 - p_e')^7 - 7p_e'(1 - p_e')^6$, where p_e' is the probability of a bit error occurring in the channel when channel coding occurs. Here, $7p_e'(1 - p_e')^6$ equals the probability of exactly one of seven bits emerging from the channel in error; The channel decoder corrects this type of error, and all data bits in the block are received correctly.

portion of **G** and attaching to it an identity matrix. For our $(7, 4)$ code,

$$\mathbf{H} = \left[\underbrace{\begin{matrix}1 & 1 & 1 & 0\\ 0 & 1 & 1 & 1\\ 1 & 1 & 0 & 1\end{matrix}}_{\text{Lower portion of } \mathbf{G}}\left|\begin{matrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{matrix}\right.\right] .$$

The parity check matrix thus has size $(N - K) \times N$, and the result of multiplying this matrix with a received word is a length-$(N - K)$ binary vector. If no digital channel errors occur—we receive a codeword so that $\widehat{\mathbf{c}} = \mathbf{c}$ —$\mathbf{H}\widehat{\mathbf{c}} = \mathbf{0}$. For example, the first column of **G**, col$[1, 0, 0, 0, 1, 0, 1]$, is a codeword. Simple calculations show that multiplying this vector by **H** results in a length-$(N - K)$ zero-valued vector.

Question 6.27 $\{263\}$ *Show that* $\mathbf{Hc} = \mathbf{0}$ *for all the columns of* $\mathbf{G}$*. Does this check guarantee that* all *codewords also have this property?*

When the received bits $\widehat{\mathbf{c}}$ do *not* form a codeword, $\mathbf{H}\widehat{\mathbf{c}}$ does not equal zero, indicating the presence of one or more errors induced by the digital channel. Note that the presence of an error can be mathematically written as $\widehat{\mathbf{c}} = \mathbf{c} \oplus \mathbf{e}$ with $\mathbf{e}$ a vector of binary values having a `1` in those positions where a bit error occurred.

Question 6.28 $\{263\}$ *Show that adding the error vector col*$[\mathtt{1}, \mathtt{0}, \dots, \mathtt{0}]$ *to a codeword flips the codeword's leading bit and leaves the rest unaffected.*

Consequently, $\mathbf{H}\widehat{\mathbf{c}} = \mathbf{H}(\mathbf{c} \oplus \mathbf{e}) = \mathbf{He}$. Because the result of the product is a length-$(N-K)$ vector of binary values, we can have $2^{N-K}-1$ non-zero values that correspond to non-zero error patterns $\mathbf{e}$. To perform our channel decoding,

1. compute (conceptually at least) $\mathbf{H}\widehat{\mathbf{c}}$;
2. if this result is zero, no detectable or correctable error occurred;
3. if non-zero, consult a table of length-$(N-K)$ binary vectors to associate them with the *minimal* error pattern that could have resulted in the non-zero result; then
4. add the error vector thus obtained to the received vector $\widehat{\mathbf{c}}$ to correct the error (because $\mathbf{c} \oplus \mathbf{e} \oplus \mathbf{e} = \mathbf{c}$).
5. Select the data bits from the corrected word to produce the received bit sequence $\widehat{b}(n)$.

The phrase *minimal* in the third item raises the point that a double (or triple or quadruple ...) error occurring during the transmission/reception of one codeword can create the same received word as a single-bit error or no error in *another* codeword. For example, col$[\mathtt{1},\mathtt{0},\mathtt{0},\mathtt{0},\mathtt{1},\mathtt{0},\mathtt{1}]$ and col$[\mathtt{0},\mathtt{1},\mathtt{0},\mathtt{0},\mathtt{1},\mathtt{1},\mathtt{1}]$ are both codewords in our example $(7,4)$ code (Eq. (6.11) $\{211\}$). The second results when the first one experiences three bit errors (first, second, and sixth bits). Such an error pattern cannot be detected by our coding strategy, but such multiple error patterns are very unlikely to occur. Our receiver uses the principle of maximum probability: An error-free transmission is much more likely than one with three errors once the bit-error probability p_e is small enough.

Question 6.29 $\{263\}$ *How small must* p_e *so that a single-bit error is more likely to occur than a triple-bit error?*

For our $(7,4)$ example, we have $2^{N-K}-1 = 7$ error patterns that can be corrected. We start with single-bit error patterns, and multiply them by the parity check matrix. If we obtain unique answers, we are done; if two or more error patterns yield the same result, we can try double-bit error patterns. In our case, single-bit error patterns give a unique result.

e	**He**
`1000000`	`101`
`0100000`	`111`
`0010000`	`110`
`0001000`	`011`
`0000100`	`100`
`0000010`	`010`
`0000001`	`001`

This corresponds to our decoding table: We associate the parity check matrix multiplication result with the error pattern and add this to the received word. If more than one error occurs (unlikely though it may be), this "error correction" strategy usually makes the error worse in the sense that more bits are changed from what was transmitted.

Note that our $(7, 4)$ code has the length and number of data bits that perfectly fits correcting single bit errors. This pleasant property arises because the number of error patterns that can be corrected, $2^{N-K} - 1$, equals the codeword length N. Codes that have $2^{N-K} - 1 = N$ are known as *Hamming codes*, and the following table provides the parameters of these codes.

N	K	E
3	1	0.33
7	4	0.57
15	11	0.73
31	26	0.84
63	57	0.90
127	120	0.94

Hamming codes are the simplest single-bit error correction codes, and the generator/parity check matrix formalism for channel coding and decoding works for them.

Unfortunately, for such large blocks, the probability of multiple-bit errors can exceed the number of single-bit errors unless the channel single-bit error probability p_e is very small. Consequently, we need to enhance the code's error correcting capability by adding double as well as single-bit error correction.

Question 6.30 {263} *What must the relation between N and K be for a code to correct all single- and double-bit errors with a "perfect fit"?*

As the block length becomes larger, more error correction will be needed. Do codes exist that can correct *all* errors? Perhaps the crowning achievement of Claude Shannon's creation of information theory answers this question. His result comes in two complementary forms: the Noisy Channel Coding Theorem and its converse.

Noisy Channel Coding Theorem Let E denote the efficiency of an error-correcting code: the ratio of the number of data bits to the total number of bits used to represent them. If the efficiency is less than the *capacity* of the digital channel, an error-correcting code exists that has the property that as the length of the code increases, the probability of an error occurring in the decoded block approaches zero.

$$\lim_{N \to \infty} \Pr[\text{block error}] = 0 \quad \text{if } E < C$$

Converse to the Noisy Channel Coding Theorem If $E > C$, the probability of an error in a decoded block must approach one regardless of the code that might be chosen.

$$\lim_{N \to \infty} \Pr[\text{block error}] = 1$$

These results mean that it is possible to transmit digital information over a noisy channel (one that introduces errors) and receive the information without error *if* the code is sufficiently *inefficient* compared to the channel's characteristics. Generally, a channel's capacity changes with the signal-to-noise ratio: As one increases or decreases so does the other. The capacity measures the overall error characteristics of a channel — the smaller the capacity the more frequently errors occur — and an overly efficient error-correcting code will not build in enough error correction capability to counteract channel errors.

This result astounded communication engineers when Shannon published it in 1948. Analog communication always yields a noisy version of the transmitted signal; in digital communication, error correction can be powerful enough to correct all errors as the block length increases. The key for this capability to exist is that the code's efficiency be less than the channel's capacity. For a binary symmetric channel, the capacity is given by

$$C = 1 + p_e \log_2 p_e + (1 - p_e) \log_2 (1 - p_e) \text{ bits/transmission.}$$

Fig. 6.15 shows how capacity varies with error probability. For example, our $(7, 4)$ Hamming code has an efficiency of 0.57, and codes having the same efficiency but longer block sizes can be used on additive noise channels where the signal-to-noise ratio exceeds 0 dB.

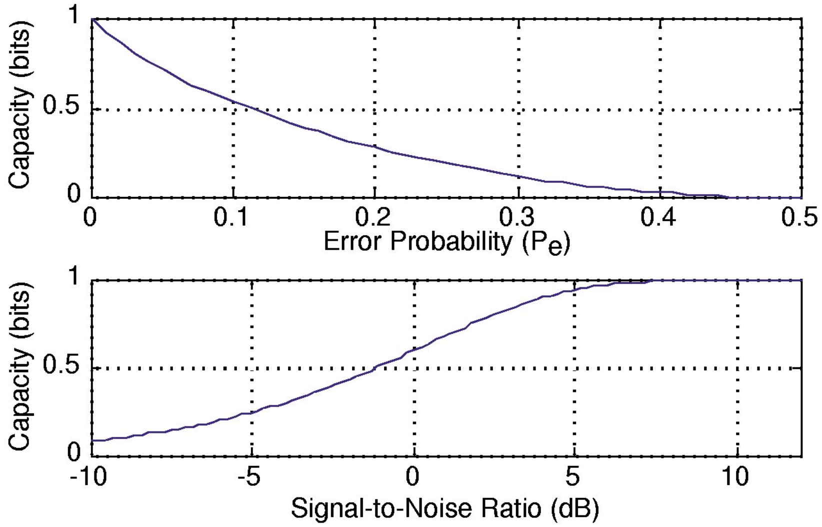


Figure 6.15 The capacity per transmission through a binary symmetric channel is plotted as a function of the digital channel's error probability (upper) and as a function of the signal-to-noise ratio for a BPSK signal set (lower).

Shannon also derived the capacity for a bandlimited (to W Hz) additive white noise channel. For this case, the signal set is unrestricted, even to the point that more than one bit can be transmitted each "bit interval." Instead of constraining channel code efficiency, the revised Noisy Channel Coding Theorem states that some error-correcting code exists such that as the block length increases, error-free transmission is possible if the source coder's datarate, $\overline{B}(\mathcal{A})R$, is less than capacity.

$$C = W \log_2(1 + \mathit{SNR}) \text{ bits/s}$$

This result proscribes the maximum datarate of the source coder's output that can be transmitted through the bandlimited channel with no error.* Shannon's proof of his theorem was very clever, and did not indicate what this code might be; it has never been found. Codes such as the Hamming code work quite well in practice to keep error rates low, but they remain greater than zero. Until the "magic" code is found, more important in communication system design is the converse. It states that if your data rate exceeds capacity, errors will overwhelm you no matter what channel coding you use. For this reason, capacity calculations are made to place limits on transmission rates.

---

*The bandwidth restriction arises not so much from channel properties, but from spectral regulation, especially for wireless channels.

Question 6.31 {**263**} *The first definition of capacity applies only for binary symmetric channels, and represents the number of bits/transmission. The second result states capacity more generally, having units of bits/second. How would you convert the first definition's result into units of bits/second?*

Example

The telephone channel has a bandwidth of 3 kHz and a signal-to-noise ratio exceeding 30 dB (at least they promise this much). The maximum data rate a modem can produce for this wireline channel and hope that errors will not become rampant is the capacity.

$$\begin{aligned} C &= 3 \times 10^3 \log_2(1 + 10^3) \\ &= 29.901 \text{ kbps} \end{aligned}$$

Thus, the so-called 33 kbps modems operate right at the capacity limit.

Note that the data rate allowed by the capacity can exceed the bandwidth when the signal-to-noise ratio exceeds 0 dB. Our results for BPSK and FSK indicated the bandwidth they require exceeds $1/T$. What kind of signal sets might be used to achieve capacity? Modem signal sets send more than one bit/transmission using a number, one of the most popular of which is *multi-level signaling*. Here, we can transmit several bits during one transmission interval by representing bit by some signal's amplitude. For example, two bits can be sent with a signal set comprised of a sinusoid with amplitudes of $\pm A$ and $\pm A/2$.

6.4 Comparison of Analog and Digital Communication

Analog communication systems, amplitude modulation (AM) radio being a typifying example, can inexpensively communicate a bandlimited analog signal from one location to another (point-to-point communication) or from one point to many (broadcast). Although we have not shown it here, the coherent receiver provides the largest possible signal-to-noise ratio for the demodulated message. Our analysis of this receiver thus indicates that some residual error will *always* be present in an analog system's output.

Although analog systems are more expensive in many circumstances than digital ones, digital systems offer much more efficiency, better performance, and much greater flexibility.

- **Efficiency**
  The Source Coding Theorem allows quantification of just how complex a

given message source is and allows us to exploit that complexity by source coding (compression). In analog communication, the only parameters of interest are message bandwidth and amplitude. We cannot exploit signal structure to achieve a more efficient communication system.

- **Performance**
  Because of the Noisy Channel Coding Theorem, we have a specific criterion by which to formulate error-correcting codes that can bring as close to error-free transmission as we might want. Even though we may send information by way of a noisy channel, digital schemes are capable of error-free transmission while analog ones cannot overcome channel disturbances; see Problem 6.7 {233} for a comparison.
- **Flexibility**
  Digital communication systems can transmit real-valued discrete-time signals, which could be analog ones obtained by analog-to-digital conversion, *and* symbolic-valued ones (computer data, for example). Any signal that can be transmitted be transmitted by analog means can be sent by digital means, with the only issue being the number of bits used in A/D conversion (how accurately do we need to represent signal amplitude). Images can be sent by analog means (commercial television), but better communication performance occurs when we use digital systems (HDTV).

  In addition to digital communication's ability to transmit a wider variety of signals than analog systems, point-to-point digital systems can be organized into global (and beyond as well) systems that provide efficient and flexible information transmission. *Computer networks*, explored in the next section, are what we call such systems today. Even analog-based networks, such as the telephone system, employ modern computer networking ideas rather than the purely analog systems of the past.

Consequently, with the increased speed of digital computers, the development of increasingly efficient algorithms, and the ability to interconnect computers to form a communications infrastructure, digital communication is now the best choice for many situations.

6.5 Communication Networks

Communication networks elaborate the Fundamental Model of Communications. The model shown in Figure 1.1 {4} describes *point-to-point* communications well, wherein the link between transmitter and receiver is straightforward, and they have the channel to themselves. One modern example of this communications mode is

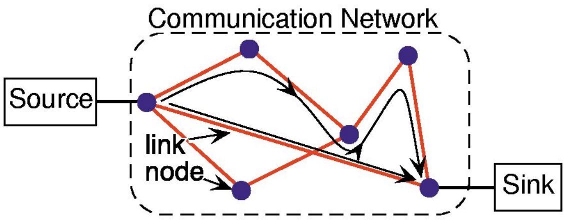


Figure 6.16 The prototypical communications network—whether it be the postal service, cellular telephone, or the Internet—consists of nodes interconnected by links. Messages formed by the source are transmitted within the network by dynamic routing. Two routes are shown. The longer one would be used if the direct link were disabled or congested.

the modem that connects a personal computer with an information server via a telephone line. The key aspect, some would say flaw, of this model is that the channel is *dedicated*: Only one communications link through the channel is allowed for all time. Regardless whether we have a wireline or wireless channel, communication bandwidth is precious, and if it could be shared without significant degradation in communications performance (measured by signal-to-noise ratio for analog signal transmission and by bit-error probability for digital transmission) so much the better.

The idea of a network first emerged with perhaps the oldest form of organized communication: the postal service. Most communication networks, even modern ones, share many of its aspects.

- A user writes a letter, serving in the communications context as the message source.
- This message is sent to the network by delivery of one of the network's public entry points. Entry points in the postal case are mailboxes, post offices, or your friendly mailman or mailwoman picking up the letter.
- The communications network delivers the message in the most efficient (timely) way possible, trying not to corrupt the message while doing so.
- The message arrives at one of the network's exit points, and is delivered to the recipient (what we have termed the message sink).

Question 6.32 {263} *Develop the network model for the telephone system, making it as analogous as possible with the postal service-communications network metaphor.*

What is most interesting about the network system is the ambivalence of the message source and sink about how the communications link is made. What they do care about is message integrity and communications efficiency. Furthermore, today's networks use heterogeneous links. Communication paths that form the Internet use wireline, optical fiber, and satellite communication links.

The first *electrical* communications network was the telegraph. Here the network consisted of telegraph operators who transmitted the message efficiently using Morse code and *routed* the message so that it took the shortest possible path to its destination while taking into account internal network failures (downed lines, drunken operators). From today's perspective, the fact that this nineteenth century system handled digital communications is astounding. Morse code, which assigned a sequence of dots and dashes to each letter of the alphabet, served as the source coding algorithm* and the signal set consisted of a short and a long pulse. Rather than a matched filter, the receiver was the operator's ear, and he wrote the message (translating from received bits to symbols).

Internally, communication networks do have point-to-point communication links between network *nodes* well described by the Fundamental Model of Communications. However, many messages share the communications channel between nodes using what we call *time-domain multiplexing*: Rather than the continuous communications mode implied in the Model as presented, message sequences are sent, sharing in time the channel's capacity. At a grander viewpoint, the network must *route* messages — decide what nodes and links to use —based on destination information — the *address* — that is usually separate from the message information. Routing in networks is necessarily dynamic: The complete route taken by messages is formed as the network handles the message, with nodes relaying the message having some notion of the best possible path at the time of transmission. Note that no omnipotent router views the network as a whole and pre-determines every message's route. Certainly in the case of the postal system dynamic routing occurs, and can consider issues like inoperative and overly busy links. In the telephone system, routing takes place when you place the call; the route is fixed once the phone starts ringing.

Modern communication networks strive to achieve the most efficient (timely) and most reliable information delivery system possible. Networks have many facets, and we explore then in turn.

---

*Because of the need for a comma between dot-dash sequences to define letter (symbol) boundaries, the average number of bits/symbol exceeded the Source Coding Theorem's *upper* bound (6.9) {203}.

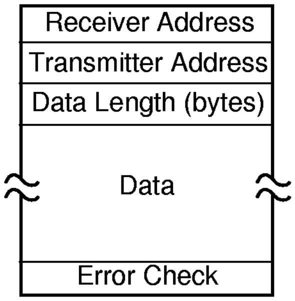


Figure 6.17 Long messages, such as files, are broken into separate packets, then transmitted over computer networks. A packet, like a letter, contains the destination address, the return address (transmitter address), and the data. The data includes the message part and a sequence number identifying its order in the transmitted message.

6.5.1 Message routing

Focusing on electrical networks, most analog ones make inefficient use of communication links because truly dynamic routing is difficult, if not impossible, to obtain. In radio networks, such as commercial television, each station has a dedicated portion of the electromagnetic spectrum, and this spectrum cannot be shared with other stations or used in any other than the regulated way. The telephone network is more dynamic, but once it establishes a call the path through the network is fixed. The users of that path control its use, and may not make efficient use of it (long pauses while one person thinks, for example). Telephone network customers would be quite upset if the telephone company momentarily disconnected the path so that someone else could use it. This kind of connection through a network —fixed for the duration of the communication session —is known as a *circuit-switched* connection.

During the 1960s, it was becoming clear that not only was digital communication technically superior, but also that the wide variety of communication modes—computer login, file transfer, and electronic mail—needed a different approach than point-to-point. The notion of computer networks was born then, and what was then called the ARPANET, now called the Internet, was born. Computer networks elaborate the basic network model by subdividing messages into smaller chunks called *packets* (Fig. 6.17). The rationale for the network enforcing smaller transmissions was that large file transfers would consume network resources all along the route, and, because of the long transmission time, a communication failure might require retransmission of the entire file. By creating packets, each of which has its own address and is routed independently of others, the network can better manage congestion. The analogy is that the postal service, rather than sending a long letter in the envelope you provide, opens the envelope, places each page in a separate envelope, and using the address on your envelope, addresses each page's envelope accordingly, and mails them separately. The network does need to make sure packet sequence (page numbering) is maintained, and the network exit

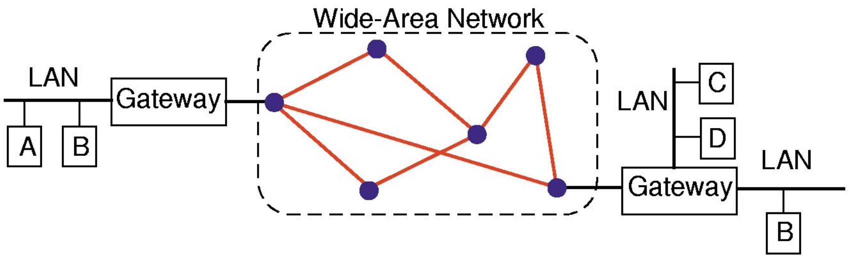


Figure 6.18 The gateway serves as an interface between local area networks and the Internet. The two shown here translate between LAN and WAN protocols; one of these also interfaces between two LANs, presumably because together the two LANs would be geographically too dispersed.

point must reassemble the original message accordingly.

Communications networks are now categorized according to whether they use packets or not. A system like the telephone network is said to be *circuit switched*: The network establishes a *fixed* route that lasts the entire duration of the message. Circuit switching has the advantage that once the route is determined, the users can use the capacity provided them however they like. Its main disadvantage is that the users may not use their capacity efficiently, clogging network links and nodes along the way. *Packet-switched* networks continuously monitor network utilization, and route messages accordingly. Thus, messages can, on the average, be delivered efficiently, but the network cannot guarantee a specific amount of capacity to the users.

6.5.2 Network architectures and interconnection

The network structure—its architecture—shown in Fig. 6.16 typifies what are known as *wide area networks* (WANs). The nodes, and users for that matter, are spread geographically over long distances. "Long" has no precise definition, and is intended to suggest that the communication links vary widely. The Internet is certainly the largest WAN, spanning the entire earth and beyond. *Local area networks*, LANs, employ a single communication link and special routing. Perhaps the best known LAN is Ethernet, which we detail subsequently. LANs connect to other LANs and to wide area networks through special nodes known as *gateways* (Fig. 6.18). In the Internet, a computer's address consists of a four byte sequence, which is known as its *IP address* (Internet Protocol address). An example address is `128.42.4.32`: each byte is separated by a period. The first two bytes specify the computer's *domain* (here Rice University). Computers are also addressed by

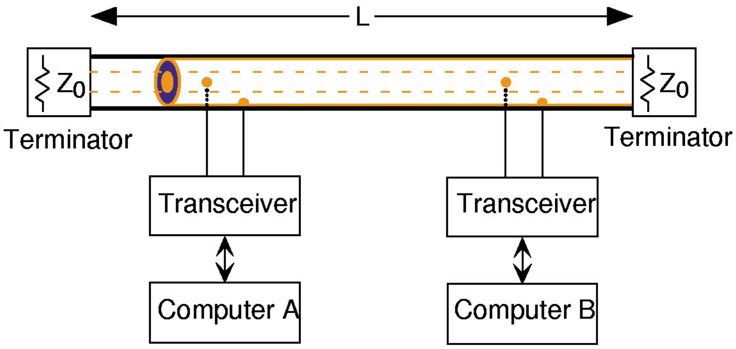


Figure 6.19 The Ethernet architecture consists of a single coaxial cable terminated at either end by a resistor having a value equal to the cable's characteristic impedance. Computers attach to the Ethernet through an interface known as a *transceiver* because it sends as well as receives bit streams represented as analog voltages.

a more human-readable form: a sequence of alphabetic abbreviations representing institution, type of institution, and computer name. A given computer has both names (`128.42.4.32` is the same as `soma.rice.edu`). Data transmission on the Internet requires the numerical form. So-called *name servers* translate between alphabetic and numerical forms, and the transmitting computer requests this translation before the message is sent to the network.

Ethernet uses as its communication medium a single length of coaxial cable (Fig. 6.19). This cable serves as the "ether", through which all digital data travel. Electrically, computers interface to the coaxial cable (Figure 3.15 {55}) through a device known as a *transceiver*. This device is capable of monitoring the voltage appearing between the core conductor and the shield as well as applying a voltage to it. Conceptually it consists of two op-amps, one applying a voltage corresponding to a bit stream (transmitting data) and another serving as an amplifier of Ethernet voltage signals (receiving data). The signal set for Ethernet resembles that shown on page {192}, with one signal the negative of the other. Computers are attached in parallel, resulting in the circuit model for Ethernet shown in Fig. 6.20.

Question 6.33 {264} *From the viewpoint of a transceiver's sending op-amp, what is the load it sees and what is the transfer function between this output voltage and some other transceiver's receiving circuit? Why should the output resistor* R_{out} *be large?*

No one computer has more authority than any other to control when and how messages are sent. Without scheduling authority, you might well wonder how one

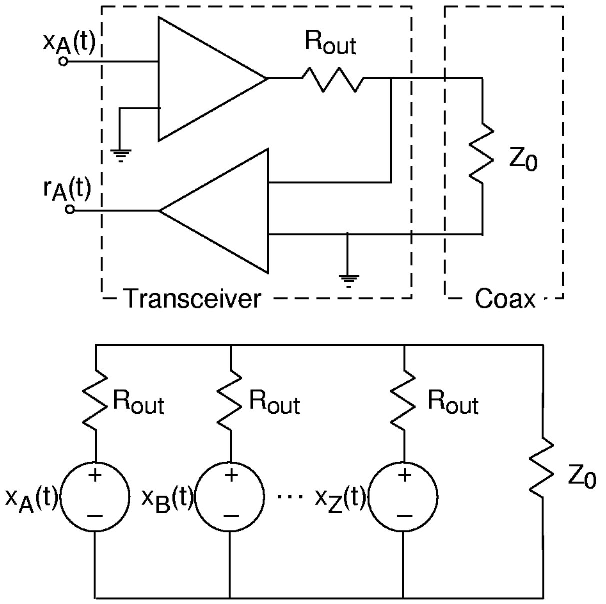


Figure 6.20 The top circuit expresses a simplified circuit model for a transceiver. The output resistance R_{out} must be much larger than Z_0 so that the sum of the various transmitter voltages add to create the Ethernet conductor-to-shield voltage that serves as the received signal $r(t)$ for all transceivers. In this case, the equivalent circuit shown in the bottom circuit applies.

---

computer sends to another without the (large) interference that the other computers would produce if they transmitted at the same time. The innovation of Ethernet is that computers schedule themselves by a *random-access* method. This method relies on the fact that *all* packets transmitted over the coaxial cable can be received by *all* transceivers, regardless of which computer the packet might actually be the intended recipient. In communications terminology, Ethernet directly supports broadcast. Each computer goes through the following steps to send a packet.

1. The computer senses the voltage across the cable to determine if some other computer is transmitting.
2. If another computer is transmitting, wait until the transmissions finish and go back to the first step. If the cable has no transmissions, begin transmitting the packet.
3. If the receiver portion of the transceiver determines that no other computer is also sending a packet, continue transmitting the packet until completion.

4. On the other hand, if the receiver senses interference from another computer's transmissions, immediately cease transmission, waiting a random amount of time to attempt the transmission again (go to step 1) until only one computer transmits and the others defer. The condition wherein two (or more) computers' transmissions interfere with others is known as a *collision*.

The reason two computers waiting to transmit may not sense the other's transmission immediately arises because of the finite propagation speed of voltage signals through the coaxial cable. The longest time any computer must wait to determine if its transmissions do not encounter interference is $2L/c$, where L is the coaxial cable's length. The maximum-length-specification for Ethernet is 1 km. Assuming a propagation speed of 2/3 the speed of light, this time interval is $10\ \mu$s. As analyzed in Problem 6.18 $\{238\}$, the number of these time intervals required, on the average, to resolve the collision is less than two!

Question 6.34 $\{$**264**$\}$ *Why does the factor of two enter into this equation? (Consider the worst-case situation of two transmitting computers located at the Ethernet's ends.)*

Thus, despite not having separate communication paths among the computers to coordinate their transmissions, the Ethernet random access protocol allows computers to communicate without only a slight degradation in efficiency, as measured by the time taken to resolve collisions relative to the time the Ethernet is used to transmit information.

A more subtle consideration in Ethernet is the minimum packet size P_{min}. The time required to transmit such packets equals P_{min}/C, where C is the Ethernet's capacity in bps. Ethernet now comes in two different types, each with individual specifications, the most distinguishing of which is capacity: 10 Mbps and 100 Mbps. If the minimum transmission time is such that packet beginning has not propagated the full length of the Ethernet before end-of-transmission, it is possible that two computers will begin transmission at the same time and, by the time their transmissions cease, the other's packet will not have propagated to the other. In this case, computers in-between the two will sense a collision, which renders both computer's transmissions senseless to them, without the two transmitting computers knowing a collision has occurred at all! For Ethernet to succeed, we must have the minimum packet transmission time exceed *twice* the voltage propagation time.

$$\frac{P_{\text{min}}}{C} > \frac{2L}{c} \quad \text{or} \quad \boxed{P_{\text{min}} > \frac{2LC}{c}}$$

Thus, for the 10 Mbps Ethernet having a 1 km maximum length specification, the minimum packet size is 200 bits.

Question 6.35 {264} *The* 100 *Mbps Ethernet was designed more recently than the* 10 *Mbps alternative. To maintain the same minimum packet size as the earlier, slower version, what should its length specification be? Why should the minimum packet size remain the same?*

6.5.3 Communication protocols

The complexity of information transmission in a computer network —reliable transmission of bits across a channel, routing, and directing information to the correct destination within the destination computer's operating system —demands an overarching concept of how to organize information delivery. No unique set of rules satisfies the various constraints communication channels and network organization place on information transmission. For example, random access issues in Ethernet are not present in wide-area networks such as the Internet. A *protocol* is a set of rules that governs how information is delivered. For example, to use the telephone network, the protocol is to pick up the phone, listen for a dial tone, dial a number having a specific number of digits, wait for the phone to ring, and say "hello." In radio, the station uses amplitude or frequency modulation with a specific carrier frequency and transmission bandwidth, and you know to turn on the radio and tune in the station. In technical terms, no one protocol or set of protocols can be used for any communication situation. Be that as it may, communication engineers have found that a common thread runs through the *organization* of the various protocols. This grand design of information transmission organization runs through all modern networks today.

What has been defined as a networking standard is a layered, hierarchical protocol organization. As shown in Fig. 6.21, protocols are organized by function and level of detail. Segregation of information transmission, manipulation, and interpretation into these categories directly affects how communication systems are organized, and what role(s) software systems fulfill. Although not thought about in this way in earlier times, this organizational structure governs the way communication engineers think about all communication systems, from radio to the Internet.

Question 6.36 {264} *How do the various aspects of establishing and maintaining a telephone conversation fit into this layered protocol organization?*

We now explicitly state whether we are working in the physical layer (signal set design, for example), the data link layer (source and channel coding), or any other layer. IP abbreviates Internet protocol, and governs gateways (how information is transmitted between networks having different internal organizations). TCP (transmission control protocol) governs how packets are transmitted through a wide-area

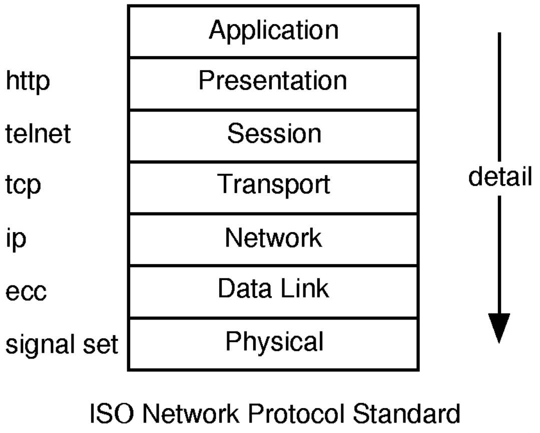


Figure 6.21 Protocols are organized according to the level of detail required for information transmission. Protocols at the lower levels (shown toward the bottom) concern reliable bit transmission. Higher level protocols concern how bits are organized to represent information, what kind of information is defined by bit sequences, what software needs the information, and how the information is to be interpreted. Bodies such as the IEEE (Institute for Electronics and Electrical Engineers) and the ISO (International Standards Organization) define standards such as this. Despite being a standard, it does not constrain protocol implementation so much that innovation and competitive individuality are ruled out.

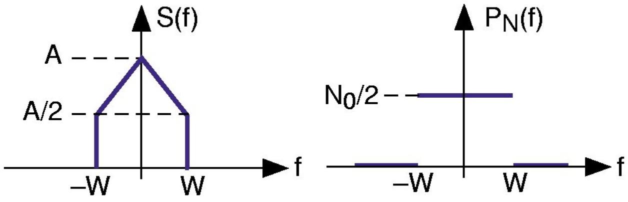
network such as the Internet. Telnet is a protocol that concerns how a person at one computer logs on to another computer across a network. A moderately high level protocol such as telnet, is not concerned with what data links (wireline or wireless) might have been used by the network or how packets are routed. Rather, it establishes connections between computers and directs each byte (presumed to represent a typed character) to the appropriate operation system component at each end. It is *not* concerned with what the characters mean or what programs the person is typing to. That aspect of information transmission is left to protocols at higher layers.

Recently, an important set of protocols created the World Wide Web. These protocols exist independently of the Internet. The Internet insures that messages are transmitted efficiently and intact; the Internet is not concerned (to date) with what messages contain. HTTP (hypertext transfer protocol) frame what messages contain and what should be done with the data. The extremely rapid development of the Web "on top" of an essentially stagnant Internet is but one example of the power of organizing how information transmission occurs without overly constraining the details.

Problems

6.1 Noise in AM Systems

The signal $\tilde{s}(t)$ emerging from an AM communication system consists of two parts: the message signal $s(t)$ and additive noise. The plot shows the message spectrum $S(f)$ and noise power spectrum $\mathcal{P}_N(f)$. The noise power spectrum lies completely within the signal's band, and has a constant value there of $\mathcal{N}_0/2$.



(a) What is the message signal's power? What is the signal-to-noise ratio?

(b) Because the power in the message decreases with frequency, the signal-to-noise ratio is not constant within subbands. What is the signal-to-noise ratio in the upper half of the frequency band?

(c) How could the system be changed so that the signal-to-noise ratio remained relatively constant across frequency *without* affecting the received message signal component?

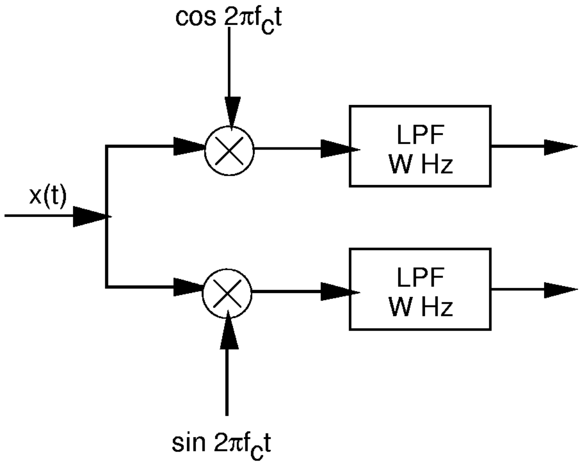
6.2 AM Stereo

Stereophonic radio transmits two signals simultaneously that correspond to what comes out of the left and right speakers of the receiving radio. While FM stereo is commonplace, AM stereo is not, but is much simpler to understand and analyze. An amazing aspect of AM stereo is that both signals are transmitted within the same bandwidth as used to transmit just one. Assume the left and right signals are bandlimited to W Hz.

$$x(t) = A[1 + s_l(t)] \cos 2\pi f_c t + A s_r(t) \sin 2\pi f_c t$$

(a) Find the Fourier transform of $x(t)$. What is the transmission bandwidth and how does it compare with that of standard AM?

(b) Let's use a coherent demodulator at the receiver. The block diagram for this receiver is

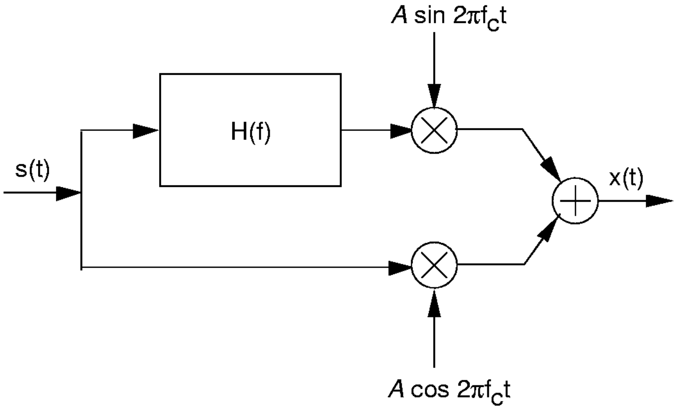


Show that this receiver indeed works: It produces the left and right signals separately.

(c) Assume the channel adds white noise to the transmitted signal. Find the signal-to-noise ratio of each signal.

6.3 A Novel Communication System

A clever system designer claims that the following transmitter has, despite its complexity, advantages over the usual amplitude modulation system. The message signal $s(t)$ is bandlimited to W Hz, and the carrier frequency $f_c \gg W$. The channel attenuates the transmitted signal $x(t)$ and adds white noise of spectral height $N_0/2$.



The transfer function $H(f)$ is given by

$$H(f) = \begin{cases} j, & f < 0 \\ -j, & f > 0 \end{cases}$$

(a) Find an expression for the spectrum of $x(t)$. Sketch your answer.

(b) Show that the usual coherent receiver demodulates this signal.

(c) Find the signal-to-noise ratio that results when this receiver is used.

(d) Find a superior receiver (one that yields a better signal-to-noise ratio), and analyze its performance.

6.4 Source Compression

Consider the following 5-letter source.

Letter	Probability
a	.5
b	.25
c	.125
d	.0625
e	.0625

(a) Find this source's entropy.

(b) Show that the simple binary coding is inefficient.

(c) Find an unequal-length codebook for this sequence that satisfies the Source Coding Theorem. Does your code achieve the entropy limit?

(d) How much more efficient is this code than the simple binary code?

6.5 Source Compression

Consider the following 5-letter source.

Letter	Probability
a	.4
b	.2
c	.15
d	.15
e	.1

(a) Find this source's entropy.

(b) Show that the simple binary coding is inefficient.

(c) Find the Huffman code for this source. What is its average code length?

6.6 Speech Compression

When we sample a signal, such as speech, we quantize the signal's amplitude to a set of integers. For a b-bit converter, signal amplitudes are represented by 2^b integers. Although these integers could be represented by a binary code for digital transmission, we should consider whether a Huffman coding would be more efficient.

(a) Load into Matlab the segment of speech contained in `y.mat`. Its sampled values lie in the interval $(-1, 1)$. To simulate a 3-bit converter, we use MATLAB's round function to create quantized amplitudes corresponding to the integers [0 1 2 3 4 5 6 7].

```
y_quant = round(3.5*y + 3.5);
```

Find the relative frequency of occurrence of quantized amplitude values. The following Matlab program computes the number of times each quantized value occurs.

```
for n=0:7
    count(n+1) = sum(y_quant == n);
end
```

Find the entropy of this source.

(b) Find the Huffman code for this source. How would you characterize this source code in words?

(c) How many fewer bits would be used in transmitting this speech segment with your Huffman code in comparison to simple binary coding?

6.7 Digital and Analog Speech Communication

Suppose we transmit speech signals over comparable digital and analog channels. We want to compare the resulting quality of the received signals. Assume the transmitters use the same power, and the channels introduce the same attenuation and additive white noise. Assume the speech signal has a 4 kHz bandwidth and, in the digital case, is sampled at an 8 kHz rate with eight-bit A/D conversion. Assume simple binary source coding and a modulated BPSK transmission scheme.

(a) What is the transmission bandwidth of the analog (AM) and digital schemes?

(b) Assume the speech signal's amplitude has a magnitude less than one. What is maximum amplitude quantization error introduced by the A/D converter?

(c) In the digital case, each bit in quantized speech sample is received in error with probability p_e that depends on signal-to-noise ratio $E_b/\mathcal{N}_0$ as depicted in the handout. However, errors in each bit have a different impact on the error in the reconstructed speech sample. Find the mean-squared error between the transmitted and received amplitude.

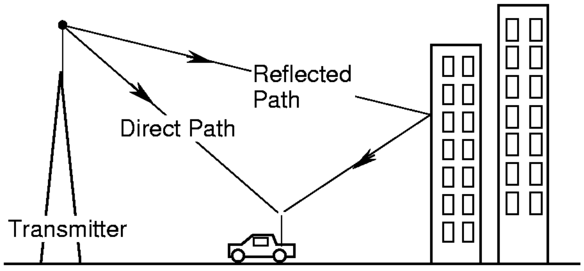
(d) In the digital case, the recovered speech signal can be considered to have two noise sources added to each sample's true value: One is the A/D amplitude quantization noise and the second is due to channel errors. Because these are separate, the total noise power equals the sum of these two. What is the signal-to-noise ratio of the received speech signal as a function of p_e?

(e) Compute and plot the received signal's signal-to-noise ratio for the two transmission schemes for a few values of channel signal-to-noise ratios.

(f) Compare and evaluate these systems.

6.8 City Radio Channels

In addition to additive white noise, metropolitan cellular radio channels also contain multipath: the attenuated signal and a delayed, further attenuated signal are received superimposed. Multipath occurs because the buildings reflect the signal and the reflected path length between transmitter and receiver is longer than the direct path.



(a) Assume that the length of the direct path is d meters and the reflected path is 1.5 times as long. What is the model for the channel, including the multipath and the additive noise?

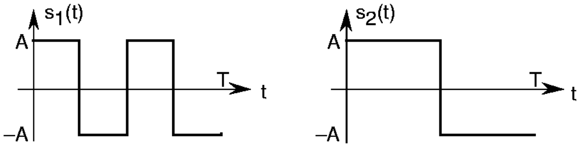
(b) Assume d is 1 km. Find and sketch the magnitude of the transfer function for the multipath component of the channel. How would you characterize this transfer function?

(c) Would the multipath affect AM radio? If not, why not; if so, how so? Would analog cellular telephone, which operates at much higher carrier frequencies (800 MHz vs. 1 MHz for radio), be affected or not? Analog cellular telephone uses amplitude modulation to transmit voice.

(d) How would the usual AM receiver be modified to minimize multipath effects? Express your modified receiver as a block diagram.

6.9 Downlink Signal Sets

In digital cellular telephone systems, the base station (transmitter) needs to relay different voice signals to several telephones at the same time. Rather than send signals at different frequencies, a clever Rice engineer suggests using a different signal set for each data stream. For example, for two simultaneous data streams, she suggests BPSK signal sets that have the depicted basic signals.



Thus, bits are represented in data stream 1 by $s_1(t)$ and $-s_1(t)$ and in data stream 2 by $s_2(t)$ and $-s_2(t)$, each of which are modulated by 900 MHz carrier. The transmitter sends the two data streams so that their bit intervals align. Each receiver uses a matched filter for its receiver. The requirement is that each receiver *not* receive the other's bit stream.

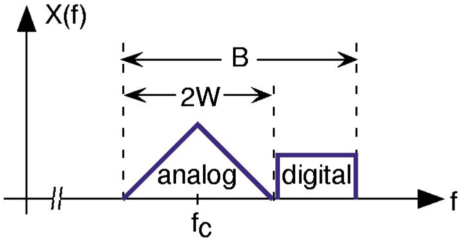
(a) What is the block diagram describing the proposed system?

(b) What is the transmission bandwidth required by the proposed system?

(c) Will the proposal work? Does the fact that the two data streams are transmitted in the same bandwidth at the same time mean that each receiver's performance is affected? Can each bit stream be received without interference from the other?

6.10 Mixed Analog and Digital Transmission

A signal $s(t)$ is transmitted using amplitude modulation in the usual way. The signal has bandwidth W Hz, and the carrier frequency is f_c. In addition to sending this analog signal, the transmitter also wants to send ASCII text in an *auxiliary band* that lies slightly above the analog transmission band. Using an 8-bit representation of the characters and a simple baseband BPSK signal set (the constant signal $+1$ corresponds to a `0`, the constant -1 to a `1`), the data signal $d(t)$ representing the text is transmitted as the same time as the analog signal $s(t)$. The transmission signal spectrum is as shown, and has a total bandwidth B.



(a) Write an expression for the time-domain version of the transmitted signal in terms of $s(t)$ and the digital signal $d(t)$.

(b) What is the maximum datarate the scheme can provide in terms of the available bandwidth?

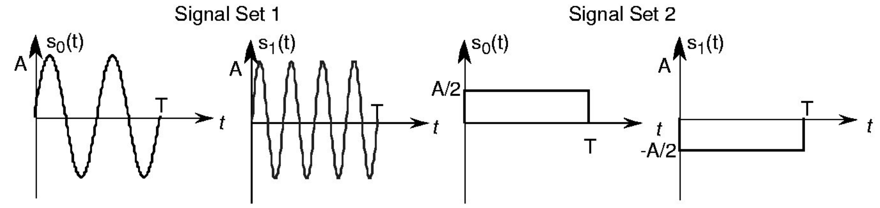
(c) Find a receiver that yields both the analog signal and the bit stream.

6.11 Digital Communication

In a digital cellular system, a signal bandlimited to 5 kHz is sampled with a two-bit A/D converter at its Nyquist frequency. The sample values are found to have the shown relative frequencies.

Sample Value	Probability
0	0.15
1	0.35
2	0.3
3	0.2

We send the bit stream consisting of Huffman-coded samples using one of the two depicted signal sets.



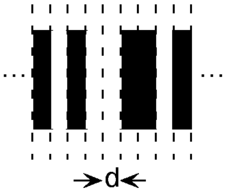
(a) What is the datarate of the compressed source?

(b) Which choice of signal set maximizes the communication system's performance?

(c) With no error-correcting coding, what signal-to-noise ratio would be needed for your chosen signal set to guarantee that the bit error probability will not exceed 10^{-3}? If the receiver moves twice as far from the transmitter (relative to the distance at which the 10^{-3} error rate was obtained), how does the performance change?

6.12 Universal Product Code

The Universal Product Code (UPC), often known as a bar code, labels virtually every sold good. An example of a portion of the code is shown.



Here a sequence of black and white bars, each having width d, presents an 11-digit number (consisting of decimal digits) that uniquely identifies the product. In retail stores, laser scanners read this code, and after accessing a database of prices, enter the price into the cash register.

(a) How many bars must be used to represent a single digit?

(b) A complication of the laser scanning system is that the bar code must be read either forwards or backwards. Now how many bars are needed to represent each digit?

(c) What is the probability that the 11-digit code is read correctly if the probability of reading a single bit incorrectly is p_e?

(d) How many error correcting bars would need to be present so that any single bar error occurring in the 11-digit code can be corrected?

6.13 Error Correcting Codes

A code maps pairs of information bits into codewords of length 5 as follows.

Data	Codeword
00	00000
01	01101
10	10111
11	11010

(a) What is this code's efficiency?

(b) Find the generator matrix **G** and parity-check matrix **H** for this code.

(c) Give the decoding table for this code. How many patterns of 1, 2, and 3 errors are correctly decoded?

(d) What is the block error probability (the probability of any number of errors occurring in the decoded codeword)?

6.14 Overly Designed Error Correction Codes

An Aggie engineer wants not only to have codewords for his data, but also to hide the information from Rice engineers (no fear of the UT engineers). He decides to represent 3-bit data with 6-bit codewords in which none of the data bits appear explicitly.

$$\begin{aligned}
c_1 &= d_1 \oplus d_2 \\
c_2 &= d_2 \oplus d_3 \\
c_3 &= d_1 \oplus d_3 \\
c_4 &= d_1 \oplus d_2 \oplus d_3 \\
c_5 &= d_1 \oplus d_2 \\
c_6 &= d_1 \oplus d_2 \oplus d_3
\end{aligned}$$

(a) Find the generator matrix **G** and parity-check matrix **H** for this code.

(b) Find a 3×6 matrix that recovers the data bits from the codeword.

(c) What is the error correcting capability of the code?

6.15 Error Correction?

It is important to realize that when more transmission errors than can be corrected, error correction algorithms believe that a smaller number of errors have occurred and correct accordingly. For example, consider a $(7,4)$ Hamming Code having the generator matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

This code corrects all single-bit error, but if a double bit error occurs, it corrects using a single-bit error correction approach.

(a) How many double-bit errors can occur in a codeword?

(b) For each double-bit error pattern, what is the result of channel decoding? Express your result as a binary error sequence for the data bits.

6.16 Selective Error Correction

We have found that digital transmission errors occur with a probability that remains constant no matter how "important" the bit may be. For example, in transmitting digitized signals, errors occur as frequently for the most significant bit and they do for the least significant bit. Yet, the former errors have a much larger impact on the overall signal-to-noise ratio than the latter. Rather than applying error correction to each sample value, why not concentrate the error correction on the most important bits? Assume that we sample an 8 kHz signal with an 8-bit A/D converter. We use single-bit error correction on the most significant four bits and none on the least significant four. Bits are transmitted using a modulated BPSK signal set over an additive white noise channel.

(a) How many error correction bits must be added to provide single-bit error correction on the most significant bits?

(b) How large must the signal-to-noise ratio of the received signal be to insure reliable communication?

(c) Assume that once error correction is applied, only the least significant 4 bits can be received in error. How much would the output signal-to-noise ratio improve using this error correction scheme?

6.17 Communication System Design
RU Communication Systems has been asked to design a communication system that meets the following requirements.

- The baseband message signal has a bandwidth of 10 kHz.
- The signal is to be sent through a noisy channel having a bandwidth of 25 kHz channel centered at 2 MHz and a signal-to-noise ratio within that band of 10 dB.
- Once received, the message signal must have a signal-to-noise ratio of at least 20 dB.

Can these specifications be met? Justify your answer.

6.18 Optimal Ethernet Random Access Protocols
Assume a population of N computers want to transmit information on a random access channel. The access algorithm works as follows.

1. Before transmitting, flip a coin that has probability p of coming up heads.
2. If only one of the N computer's coins comes up heads, its transmission occurs successfully, and the others must wait until that transmission is complete and then resume the algorithm.
3. If none or more than one head comes up, the N computers will either remain silent (no heads) or a collision will occur (more than one head). This unsuccessful transmission situation will be detected by all computers once the signals have propagated the length of the cable, and the algorithm resumes (go to step 1).

(a) What is the optimal probability to use for flipping the coin? In other words, what should p be to maximize the probability that exactly one computer transmits?

(b) What is the probability of one computer transmitting when this optimal value of p is used as the number of computers grows to infinity?

(c) Using this optimal probability, what is the average number of coin flips that will be necessary to resolve the access so that one computer successfully transmits?

(d) Evaluate this algorithm. Is it realistic? Is it efficient?

Appendix A

Complex Numbers

A.1 Definitions

The notion of the square root of -1 originated with the quadratic formula: The solution of certain quadratic equations exist only if the so-called imaginary quantity $\sqrt{-1}$ could be defined. Mathematicians and physicists use the symbol i for $\sqrt{-1}$. Because electrical engineers have historically used this symbol to represent current, we use the symbol j.

A *complex number* z has the form $a + jb$, where a,b are real numbers. This representation is known as the Cartesian form of z. The *real part* of a complex number z, written as $\mathrm{Re}\,[z]$, equals a. We consider the real part as a function that works by selecting that component of a complex number *not* multiplied by j. The *imaginary part* of z, $\mathrm{Im}\,[z]$, equals b: That part of a complex number that is multipled by j. Note that both the real and imaginary parts of a complex number are real-valued.

The *complex conjugate* of z, written as z^*, has the same real part as z but an imaginary part of the opposite sign.

$$\begin{aligned} z &= \mathrm{Re}\,[z] + j\mathrm{Im}\,[z] \\ z^* &= \mathrm{Re}\,[z] - j\mathrm{Im}\,[z] \end{aligned}$$

Complex numbers can also be expressed in an alternate form, *polar form*, which we will find quite useful. Polar form arises because of Euler's relations.

$$\boxed{\begin{aligned} e^{j\theta} &= \cos\theta + j\sin\theta \\ \cos\theta &= \frac{e^{j\theta}+e^{-j\theta}}{2} \\ \sin\theta &= \frac{e^{j\theta}-e^{-j\theta}}{2j} \end{aligned}}$$

The first of these is easily derived from the Taylor's series for the exponential.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Substituting $j\theta$ for x, we find that

$$e^{j\theta} = 1 + j\frac{\theta}{1!} - \frac{\theta^2}{2!} - j\frac{\theta^3}{3!} + \cdots$$

because $j^2 = -1$, $j^3 = -j$, and $j^4 = 1$. Grouping separately the real-valued terms and the imaginary-valued ones,

$$e^{j\theta} = 1 - \frac{\theta^2}{2!} + \cdots + j\left(\frac{\theta}{1!} - \frac{\theta^3}{3!} + \cdots\right)$$

The real-valued terms correspond to the Taylor's series for $\cos\theta$, the imaginary ones to $\sin\theta$, and Euler's first relation results. The remaining relations are easily derived from the first. Returning to the Cartesian form for a complex number,

$$a + jb = \sqrt{a^2+b^2}\left(\frac{a}{\sqrt{a^2+b^2}} + j\frac{b}{\sqrt{a^2+b^2}}\right)$$

By forming a right triangle having sides a and b, we see that the terms inside the parentheses can be a cosine and a sine. We thus obtain the *polar form* for complex numbers.

$$\boxed{z = a + jb = re^{j\theta} \qquad \begin{array}{ll} a = r\cos\theta & r = |z| = \sqrt{a^2+b^2} \\ b = r\sin\theta & \theta = \tan^{-1}\dfrac{b}{a} \end{array}}$$

The quantity r is known as the *magnitude* of the complex number z, and is frequently written as $|z|$. The quantity θ is the complex number's *angle*. In using the arc tangent formula to find the angle, we must take into account the quadrant in which the complex number lies. We can express the polar form for z as $r\angle\theta$, but computationally, we must use the exponential form $re^{j\theta}$.

Question A.1 {**264**} *Convert* $3 - 2j$ *to polar form.*

A.2 Arithmetic in Cartesian Form

Adding and subtracting complex numbers expressed in Cartesian form is quite easy: You add (subtract) the real parts and imaginary parts separately.

$$z_1 \pm z_2 = a_1 \pm a_2 + j(b_1 \pm b_2)$$

Question A.2 {**264**} *Use this defintion of addition to show that the real and imaginary part functions can be expressed as a combination of a complex number and its conjugate: Re* $[z] = (z + z^*)/2$ *and Im* $[z] = (z - z^*)/2j$.

To multiply two complex numbers in Cartesian form is not quite as easy.

$$z_1 z_2 = (a_1 + jb_1)(a_2 + jb_2) = a_1 a_2 - b_1 b_2 + j(a_1 b_2 + a_2 b_1)$$

Question A.3 {**264**} *What is the product of a complex number and its conjugate?*

Division requires mathematical manipulation. We convert the division problem into a multiplication problem by multiplying both the numerator and denominator by the conjugate of the denominator.

$$\frac{z_1}{z_2} = \frac{a_1 + jb_1}{a_2 + jb_2} = \frac{(a_1 + jb_1)(a_2 - jb_2)}{a_2^2 + b_2^2} = \frac{a_1 a_2 + b_1 b_2 + j(a_2 b_1 - a_1 b_2)}{a_2^2 + b_2^2}$$

Because the final result is so complicated, it's best to remember *how* to perform division—multiplying numerator and denominator by the complex conjugate of the denominator—than to try to remember the final result.

Question A.4 {**265**} *Evaluate the ratio* $(1 + j)/(1 - j)$.

A.3 Arithmetic in Polar Form

Because of the properties of the exponential, multiplication and division of two complex numbers is more easily computed when they are expressed in polar form.

$$\boxed{\begin{aligned} z_1 z_2 &= (r_1 e^{j\theta_1})(r_2 e^{j\theta_2}) = r_1 r_2 e^{j(\theta_1 + \theta_2)} \\ \frac{z_1}{z_2} &= \frac{r_1 e^{j\theta_1}}{r_2 e^{j\theta_2}} = \frac{r_1}{r_2} e^{j(\theta_1 - \theta_2)} \end{aligned}}$$

To multiply, the radius equals the product of the radii and the angle the sum of the angles. To divide, the radius equals the ratio of the radii and the angle the difference of the angles. When the original complex numbers are in Cartesian form, it's usually worth translating into polar form, then performing the multiplication or division (especially in the case of the latter). Addition and subtraction of polar forms amounts to converting to Cartesian form, performing the arithmetic operation, and converting back to polar form.

Example

When we solve circuit problems, the crucial quantity, known as a transfer function, will always be expressed as the ratio of polynomials in the variable $s = j2\pi f$. What we'll need to understand the circuit's effect is the transfer function in polar form. For instance, suppose the transfer function equals

$$\frac{s+2}{s^2+s+1}, \quad s = j2\pi f\ .$$

Performing the required division is most easily accomplished by first expressing the numerator and denominator each in polar form, then calculating the ratio. Thus,

$$\begin{aligned}
\left.\frac{s+2}{s^2+s+1}\right|_{s=j2\pi f} &= \frac{j2\pi f+2}{-4\pi^2 f^2 + j2\pi f + 1} \\
&= \frac{\sqrt{4+4\pi^2 f^2}\, e^{j\tan^{-1}\pi f}}{\sqrt{(1-4\pi^2 f^2)^2 + 4\pi^2 f^2}\, e^{j\tan^{-1} 2\pi f/(1-4\pi^2 f^2)}} \\
&= \sqrt{\frac{4+4\pi^2 f^2}{1-4\pi^2 f^2 + 16\pi^4 f^4}}\, e^{j[\tan^{-1}\pi f - \tan^{-1} 2\pi f/(1-4\pi^2 f^2)]}
\end{aligned}$$

Problems

A.1 Convert the following expressions into polar form. Plot their location in the complex plane.

(a) $(1+\sqrt{-3})^2$ **(b)** $3+j^4$

(c) $\dfrac{2-j6/\sqrt{3}}{2+j6/\sqrt{3}}$ **(d)** $(4-j^3)(1+j\frac{1}{2})$

(e) $3e^{j\pi}+4e^{j\pi/2}$ **(f)** $(\sqrt{3}+j)2\sqrt{2}e^{-j\pi/4}$

(g) $\dfrac{3}{1+j3\pi}$

Appendix B

Decibels

The decibel scale expresses amplitudes and power values *logarthimically.* The definitions for these differ, but are consistent with each other.

$$\begin{aligned}\text{power}[s] \text{ in decibels} &= 10\log_{10}\frac{\text{power}[s]}{\text{power}[s_0]}\\ \text{amplitude}[s] \text{ in decibels} &= 20\log_{10}\frac{\text{amplitude}[s]}{\text{amplitude}[s_0]}\end{aligned}$$

Here $\text{power}[s_0]$ and $\text{amplitude}[s_0]$ represent a *reference* power and amplitude, respectively. Quantifying power or amplitude in decibels essentially means that we are comparing quantities to a standard or that we want to express how they changed. You will hear statements like "The signal went down by 3 dB" and "The filter's gain in the stopband is -60 dB." (Decibels is abbreviated to dB.)

Question B.1 $\{$**265**$\}$ *The prefix "deci" implies a tenth; a decibel is a tenth of a Bel. Who is this measure named for?*

The consistency of these two definitions arises because power is proportional to the square of amplitude: $\text{power}[s] \propto (\text{amplitude}[s])^2$. Plugging this expression into the definition for decibels, we find that

$$\begin{aligned}10\log_{10}\frac{\text{power}[s]}{\text{power}[s_0]} &= 10\log_{10}\frac{(\text{amplitude}[s])^2}{(\text{amplitude}[s_0])^2}\\ &= 20\log_{10}\frac{\text{amplitude}[s]}{\text{amplitude}[s_0]}\,.\end{aligned}$$

Because of this consistency, *stating relative change in terms of decibels is unambiguous.* A factor of 10 increase in amplitude corresponds to a 20 dB increase in both amplitude and power!

Power Ratio	dB
1	0
$\sqrt{2}$	1.5
2	3
$\sqrt{10}$	5
4	6
5	7
8	9
10	10
0.1	−10

Table B.1 Common values for the decibel. The decibel values for all but the powers of ten are approximate, but are accurate to a decimal place.

The accompanying table provides "nice" decibel values. Converting decibel values back and forth is fun, and tests your ability to think of decibel values as sums and/or differences of the well-known values and of ratios as products and/or quotients. This conversion rests on the logarithmic nature of the decibel scale. For example, to find the decibel value for $\sqrt{2}$, we halve the decibel value for 2; 26 dB equals $10 + 10 + 6$ dB that corresponds to a ratio of $10 \cdot 10 \cdot 4 = 400$. Decibel quantities add; ratio values multiply.

One reason decibels are used so much is the frequency-domain input-output relation for linear systems: $Y(f) = X(f) \cdot H(f)$ {109}. Because the transfer function multiplies the input signal's spectrum, to find the output amplitude at a given frequency we simply add the filter's gain in decibels (relative to a reference of one) to the input amplitude at that frequency. This calculaton is one reason that we plot transfer function magnitude on a logarithmic vertical scale expressed in decibels.

Problems

B.1 Decibel Conversions

Find the power ratios that correspond to the following decibel values. Don't use a calculator (you're on a desert island).

(a) 8 **(b)** −5 **(c)** 42 **(d)** −16 **(e)** −60 **(f)** 18

B.2 Ratio Conversions

Convert the following power ratios to decibels.

(a) 16 **(b)** 0.001 **(c)** 40
(d) 0.16 **(e)** 1.6 **(f)** 10% increase

Appendix C

Frequency Allocations

To prevent radio stations from transmitting signals "on top of each other," the United States and other national governments began in the 1930s regulating the carrier frequencies and power outputs stations could use. With increased use of the radio spectrum for both public and private use, this regulation has become increasingly important. On the next page is the so-called *Frequency Allocation Chart*, which shows what kinds of broadcasting can occur in which frequency bands. For the color, online version, see the course Web site. Detailed radio carrier frequency assignments are much too detailed to present here.

Answers to Questions

1.1 {8}

As $\cos(\alpha+\beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$, $\cos 2\pi(f+1)n = \cos 2\pi fn \cos 2\pi n - \sin 2\pi fn \sin 2\pi n = \cos 2\pi fn$.

1.2 {8}

A square wave takes on the values 1 and -1 alternately. See the plot in problem 1.1.

2.1 {17}

In the first case, order does not matter; in the second it does. "Delay" means $t \to t - \tau$. "Time-reverse" means $t \to -t$.

Case 1 $y(t) = G \cdot x(t-\tau)$, and the way we apply the gain and delay the signal gives the same result.

Case 2

Time-reverse then delay: $y(t) = x(-(t-\tau)) = x(-t+\tau)$.

Delay then time-reverse: $y(t) = x(-t-\tau)$.

2.2 {20}

$\mathrm{sq}(t) = \sum_{n=-\infty}^{\infty} (-1)^n A\, \mathrm{p}_{T/2}(t - nT/2)$

3.1 {28}

KCL says that the sum of currents entering or leaving a node must be zero. If we consider two nodes together as a "supernode," KCL applies as well to currents entering the combination. Since no currents enter an entire circuit, the sum of currents must be zero. If we had a two-node circuit, the KCL equation of one *must* be the negative of the other, We can combine all but one node in a circuit into a supernode; KCL for the supernode must be the negative of the remaining node's KCL equation. Consequently, specifying $n-1$ KCL equations always specifies the remaining one.

3.2 {29}

The circuit serves as an amplifier having a gain of $R_2/(R_1+R_2)$.

3.3 {31}

Replacing the current source by a voltage source does not change the fact that the voltages are identical. Consequently, $v_{\text{in}} = R_2 i_{\text{out}}$ or $i_{\text{out}} = v_{\text{in}}/R_2$. This result does not depend on the resistor R_1, which means that we simply have a resistor (R_2) across a voltage source. The two-resistor circuit has no apparent use.

3.8 {42}

Division by $j2\pi f$ arises from integrating a complex exponential. Consequently,

$$\frac{1}{j2\pi f}V \longleftrightarrow \int V e^{j2\pi f t}\, dt \,.$$

3.4 {32}

In a series combination of resistors, the current is the same in each; in a parallel combination, the voltage is the same. For a series combination, the equivalent resistance is the sum of the resistances, which will be larger than any component resistor's value; for a parallel combination, the equivalent conductance is the sum of the component conductances, which is larger than any component conductance. The equivalent resistance is therefore smaller than any component resistance.

3.5 {34}

$R_{\text{eq}} = R_2/(1 + R_2/R_L)$. Thus, a 10% change means that the ratio R_2/R_L must be less than 0.1. A 1% change means that $R_2/R_L < 0.01$.

3.6 {37}

$v_{\text{oc}} = R_2/(R_1+R_2)v_{\text{in}}$. $i_{\text{sc}} = -v_{\text{in}}/R_1$ (resistor R_2 is shorted out in this case). Thus, $v_{\text{eq}} = R_2/(R_1+R_2)v_{\text{in}}$ and $R_{\text{eq}} = R_1R_2/(R_1+R_2)$.

3.7 {38}

$i_{\text{eq}} = R_1/(R_1+R_2)\cdot i_{\text{in}}$ and $R_{\text{eq}} = R_3 \| (R_1+R_2)$.

3.9 {45}

The key notion is writing the imaginary part as the difference between a complex exponential and its complex conjugate:

$$\mathrm{Im}\left[Ve^{j2\pi ft}\right] = \frac{Ve^{j2\pi ft} - V^*e^{-j2\pi ft}}{2j} .$$

The response to $Ve^{j2\pi ft}$ is $VH(f)e^{j2\pi ft}$, which means the response to $V^*e^{-j2\pi ft}$ is $V^*H(-f)e^{-j2\pi ft}$. As $H(-f) = H^*(f)$, the Superposition Principle says that the output to the imaginary part is Im $\left[VH(f)e^{j2\pi ft}\right]$. The same argument holds for the real part: Re $\left[Ve^{j2\pi ft}\right] \rightarrow$ Re $\left[VH(f)e^{j2\pi ft}\right]$.

3.10 {53}

Not necessarily, especially if we desire an individual knobs for adjusting the gain and the cutoff frequency.

3.11 {59}

In both cases, the answer depends less on geometry than on material properties. For coaxial cable, $c = 1/\sqrt{\mu_d \epsilon_d}$. For twisted pair,

$$c = \frac{1}{\sqrt{\mu\epsilon}}\sqrt{\frac{\cosh^{-1}(d/2r)}{\delta/2r + \cosh^{-1}(d/2r)}} .$$

3.12 {60}

You can find these frequencies from the spectrum allocation chart Appendix C. Light in the middle of the visible band has a wavelength of about 600 nm, which corresponds to a frequency of $5/times10^{14}$ Hz. Cable television transmits within the same frequency band as broadcast television (about 200 MHz or 2×10^8 Hz. Thus, the visible electromagnetic frequencies are over six orders of magnitude higher!

3.13 {60}

As frequency increases, $2\pi f\tilde{C} \gg \tilde{G}$ and $2\pi f\tilde{L} \gg \tilde{R}$. In this high-frequency region,

$$
\begin{aligned}
\gamma &= j2\pi f\sqrt{\widetilde{L}\widetilde{C}}\sqrt{\left(1+\frac{\widetilde{G}}{j2\pi f\widetilde{C}}\right)\left(1+\frac{\widetilde{R}}{j2\pi f\widetilde{L}}\right)} \\
&\approx j2\pi f\sqrt{\widetilde{L}\widetilde{C}}\left(1+\frac{1}{2}\frac{1}{j2\pi f}\left[\frac{\widetilde{G}}{\widetilde{C}}+\frac{\widetilde{R}}{\widetilde{L}}\right]\right) \\
&\approx j2\pi f\sqrt{\widetilde{L}\widetilde{C}}+\frac{1}{2}\left(\widetilde{G}\sqrt{\frac{\widetilde{L}}{\widetilde{C}}}+\widetilde{R}\sqrt{\frac{\widetilde{C}}{\widetilde{L}}}\right)
\end{aligned}
$$

Thus, the attenuation (space) constant equals the real part of this expression, and equals $a(f) = (\widetilde{G}Z_0 + \widetilde{R}/Z_0)/2$.

3.14 {68}

The ratio between adjacent values is about $\sqrt{2}$.

4.1 {85}

The average of a set of numbers is the sum divided by the number of terms. Viewing signal integration as the limit of a Riemann sum, the integral corresponds to the average.

4.2 {88}

The rms value of a sinusoid equals its amplitude divided by $\sqrt{2}$. As a half-wave rectified sine wave is zero during half of the period, its rms value is $A/2\sqrt{2}$.

4.3 {89}

Total harmonic distortion equals

$$
\frac{\sum_{k=2}^{\infty}\left(a_k^2+b_k^2\right)}{a_1^2+b_1^2}\,.
$$

Clearly, this quantity is most easily computed in the frequency domain. However, the numerator equals the the square of the signal's rms value minus the power in the average and the power in the first harmonic.

4.4 {92}

Total harmonic distortion in the square wave is $1-(\frac{4}{\pi})^2/2 = 20\%$.

4.5 {97}

$c_0 = A\Delta/T$. This quantity clearly corresponds to the periodic pulse signal's average value.

4.6 {97}

A half-wave rectified sine wave occurs when $\Delta = T/2$. Thus, $\angle c_k = -\pi k/2 - \pi \text{neg} \frac{\sin \pi k/2}{\pi k}$, which corresponds to a linear function of k. The linear phase can be removed by delaying the signal by $1/4$ of a period.

4.7 {99}

For N signals directly encoded requires a bandwidth of N/T. Using a binary representation, we need $\log_2 N/period$. For $N = 128$, the binary-encoding scheme has a factor of $7/128 = 0.05$ smaller bandwidth. Clearly, binary encoding is superior.

4.8 {99}

We can use N different amplitude values at only one frequency to represent the various letters.

4.9 {102}

Because the filter's gain at zero frequency equals one, the average output values equal the respective average input values.

4.10 {105}

$$\begin{aligned}\mathcal{F}[S(f)] &= \int_{-\infty}^{\infty} S(f) e^{-j2\pi f t}\, df \\ &= \int_{-\infty}^{\infty} S(f) e^{+j2\pi f(-t)}\, df \\ &= s(-t)\end{aligned}$$

4.11 {105}

$\mathcal{F}[\mathcal{F}[\mathcal{F}[\mathcal{F}[s(t)]]]] = s(t)$. From the answer to the previous question (4.10), two

Fourier transforms applied to $s(t)$ yields $s(-t)$. We need two more to get us back where we started.

4.12 {108}

The signal is the inverse Fourier transform of the triangularly shaped spectrum, and equals

$$s(t) = W\left(\frac{\sin \pi W t}{\pi W t}\right)^2 .$$

4.13 {108}

The result is most easily found in the spectrum's formula: the power in the signal-related part of $x(t)$ is half the power of the signal $s(t)$.

4.14 {110}

The inverse transform of the frequency response is $\frac{1}{RC}e^{-t/RC}\,\mathrm{u}(t)$. Multiplying the frequency response by $\left[1 - e^{-j2\pi f\Delta}\right]$ means subtract from the original signal its time-delayed version Delaying the frequency response's time-domain version by Δ results in $\frac{1}{RC}e^{-(t-\Delta)/RC}\,\mathrm{u}(t-\Delta)$. Subtracting from the undelayed signal yields $\frac{1}{RC}e^{-t/RC}\,\mathrm{u}(t) - \frac{1}{RC}e^{-(t-\Delta)/RC}\,\mathrm{u}(t-\Delta)$. Now we integrate this sum. Because the integral of a sum equals the sum of the component integrals (integration is linear), we can consider each separately. Because integration and signal-delay are linear, the integral of a delayed signal equals the delayed version of the integral. The integral is provided in the example {110}.

4.15 {112}

If the glottis were linear, a constant input (a zero-frequency sinusoid) should yield a constant output. The periodic output indicates nonlinear behavior.

4.16 {116}

In the bottom-left panel, the period is about 0.009 s, which equals a frequency of 111 Hz. The bottom-right panel has a period of about 0.0065 s, a frequency of 154 Hz.

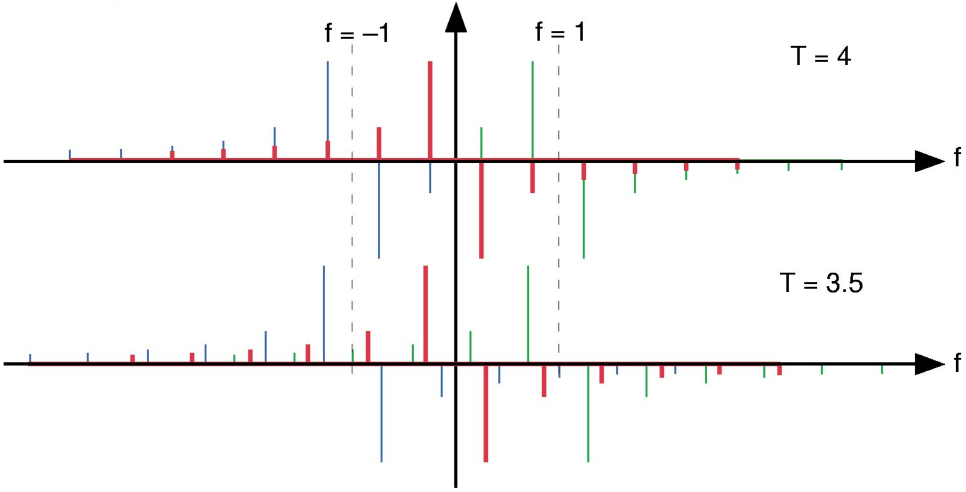
4.17 {116}

Because males have a lower pitch frequency, the spacing between spectral lines is smaller. This closer spacing more accurately reveals the formant structure. Doubling the pitch frequency to 300 Hz for Figure 4.14 {117} would amount to removing every other spectral line.

4.18 {121}

The only effect of pulse duration is to unequally weight the spectral repetitions. Because we are only concerned with the repetition centered about the origin, the pulse duration has no significant effect on recovering a signal from its samples.

4.19 {121}



The square wave's spectrum is shown by the bolder set of lines centered about the origin. The dashed lines correspond to the frequencies about which the spectral repetitions (due to sampling with $T_s = 1$) occur. As the square wave's period decreases, the negative frequency lines move to the left and the positive frequency ones to the right.

4.20 {122}

The simplest bandlimited signal is the sine wave. At the Nyquist frequency, exactly two samples/period would occur. Reducing the sampling rate would result in fewer samples/period, and these samples would appear to have arisen from a lower frequency sinusoid.

4.21 {122}

The plotted temperatures were quantized to the nearest degree. Thus, the high temperature's amplitude was quantized as a form of A/D conversion.

4.22 {123}
Solving $2^{-B} = .001$ results in $B = 10$ bits.

5.1 {135}
$25_{10} = 11001_2$ and $7_{10} = 111_2$. We find that $11001_2 + 111_2 = 100000_2 = 32_{10}$.

5.2 {136}
For b-bit signed integers, the largest number is $2^{b-1} - 1$. For $b = 32$, we have $2,147,483,647$ and for $b = 64$, we have $9,223,372,036,854,775,807$ or about 9.2×10^{18}.

5.3 {136}
In floating point, the number of bits in the exponent determines the largest and smallest representable numbers. For 32-bit floating point, the largest (smallest) numbers are $2^{\pm 127} = 1.7 \times 10^{38} (5.9 \times 10^{-39})$. For 64-bit floating point, the largest number is about 10^{9863}.

5.4 {137}
If the base is a multiple of three, then $1/3$ will have a finite expansion. For example, in base-3, $1/3 = 0.1_3$, and in base-6, $1/3 = 0.2_6$.

5.5 {137}

$$\begin{aligned} S\left(e^{j2\pi(f+1)}\right) &= \sum_n s(n) e^{-j2\pi(f+1)n} \\ &= \sum_n e^{-j2\pi n} s(n) e^{j2\pi f n} \\ &= \sum_n s(n) e^{j2\pi f n} \\ &= S\left(e^{j2\pi f}\right) \end{aligned}$$

5.6 {141}

$$\alpha \sum_{n=n_0}^{N+n_0-1} \alpha^n - \sum_{n=n_0}^{N+n_0-1} \alpha^n = \alpha^{N+n_0} - \alpha^{n_0}$$

which, after manipulation, yields the geometric sum formula.

5.7 {143}

If the sampling frequency exceeds the Nyquist frequency, the spectrum of the samples equals the analog spectrum, but over the normalized analog frequency fT. Thus, the energy in the sampled signal equals the original signal's energy multipled by T.

5.8 {145}

This situation amounts to aliasing in the time-domain.

{255}

When the signal is real-valued, we may only need half the spectral values, but the complexity remains unchanged. If the data are complex-valued, which demands retaining all frequency values, the complexity is again the same. When only K frequencies are needed, the complexity is $O(KN)$.

5.10 {147}

If a DFT required 1 ms to compute, and signal having ten times the duration would require 100 ms to compute. Using the FFT, a 1 ms computing time would increase by a factor of about $\log_2 10 = 3.3$, a factor of 30 less than the DFT would have needed.

5.11 {150}

The upper panel has not used the FFT algorithm to compute the length-4 DFTs while the lower one has. The ordering is determined by the algorithm.

5.12 {151}

Number of samples equals $1.2 \times 11,025 = 13,230$. The datarate is $11,025 \times 16 = 176.4$ kbps. The storage required would be $26,460$ bytes.

5.13 {152}
The oscillations are due to the boxcar window's Fourier transform, which equals the sinc function.

5.14 {153}
These numbers are powers-of-two, and the FFT algorithm can be exploited with these lengths. To compute a longer transform than the input signal's duration, we simply zero-pad the signal.

5.15 {154}
In discrete-time signal processing, an amplifier amounts to a multiplication, a very easy operation to perform.

5.16 {157}
Such terms would require the system to know what future input or output values would be before the current value was computed. Thus, such terms can cause difficulties.

5.17 {158}
The indices can be negative, and this condition is not allowed in MATLAB. To fix it, we must start the signals later in the array.

5.18 {160}
It now acts like a bandpass filter.

5.19 {161}
We have $p+q$ multiplications and $p-1+q-1$ additions. Thus, the total number of arithmetic operations equals $2(p+q-1)$.

5.20 {162}
The DTFT of the unit sample equals a constant (equaling 1). In this case, the Fourier transform of the output equals the transfer function.

5.21 {163}
In sampling a discrete-time signal's Fourier transform L times equally over $[0, 2\pi)$ to form the DFT, the corresponding signal equals the periodic repetition of the

original signal.

$$S(k) \longleftrightarrow \sum_{i=-\infty}^{\infty} s(n-iL)$$

To avoid aliasing (in the time domain), the transform length must equal or exceed the signal's duration.

5.22 {163}

The difference equation for an FIR filter has the form

$$y(n) = \sum_{m=0}^{q} b_m x(n-m) \; .$$

The unit-sample response equals

$$h(n) = \sum_{m=0}^{q} b_m \delta(n-m) \; ,$$

which corresponds to the representation described in Problem 5.2 {173} of a length $q+1$ signal.

5.23 {163}

The unit-sample response's duration is $q+1$ and the signal's N_x. Thus the statement is correct.

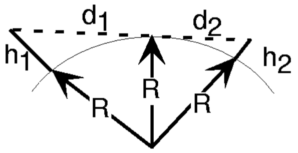
5.24 {168}

Let N denote the input's total duration. The time-domain implementation requires a total of $N(2q_1)$ computations, or $2q+1$ computations per input value. In the frequency domain, we split the input into N/N_x sections, each of which requires $\left(1+\frac{q}{N_x}\right)\log_2(N_x+q)+7\frac{q}{N_x}+6$ per input in the section. Because we divide *again* by N_x to find the number of computations per input value in the entire input, this quantity *decreases* as N_x increases. For the time-domain implementation, it stays constant.

5.25 {169}

The delay is *not* computational delay here —the plot shows the first output value is aligned with the filter's first input —although in real systems this is an important consideration. Rather, the delay is due to the filter's phase shift: A phase-shifted sinusoid is equivalent to a time-delayed one: $\cos(2\pi fn - \phi) = \cos\big(2\pi f(n - \phi/2\pi f)\big)$. All filters have phase shifts. This delay could be removed if the filter introduced no phase shift. Such filters do not exist in analog form, but digital ones can be programmed, but not in real time. Doing so would require the output to emerge before the input arrives!

6.1 {181}



Using the Pythagorean Theorem, $(h + R)^2 = R^2 + d^2$, where h is the antenna height, d is the distance from the top of the earth to a tangency point with the earth's surface, and R the earth's radius. The line-of-sight distance between two earth-based antennae equals

$$d_{\text{LOS}} = \sqrt{2h_1R + h_1^2} + \sqrt{2h_2R + h_2^2}\ .$$

As the earth's radius is much larger than the antenna height, we have to good approximation that $d_{\text{LOS}} = \sqrt{2h_1R} + \sqrt{2h_2R}$. If one antenna is at ground elevation, say $h_2 = 0$, the other antenna's range is $\sqrt{2h_1R}$.

6.2 {181}

As frequency decreases, wavelength increases and can approach the distance between the earth's surface and the ionosphere. Assuming a distance between the two of 80 km, the relation $\lambda f = c$ gives a corresponding frequency of 3.75 kHz. Such low carrier frequencies would be limited to low bandwidth analog communication and to low datarate digital communications. The US Navy did use such a communication scheme to reach all of its submarines at once.

6.3 {182}

Transmission to the satellite, known as the uplink, encounters inverse-square law power losses. Reflecting off the ionosphere not only encounters the same loss, but twice. Reflecton is the same as transmitting exactly what arrives, which means that the total loss is the *product* of the uplink and downlink losses. The geosynchronous orbit lies at an altitude of 3.57×10^7 m. The ionosphere begins at an altitude of about 50 km (5×10^4 m). The amplitude loss in the satellite case is proportional to 2.8×10^{-8}; for Marconi, it was proportional to 4×10^{-10}. Marconi was *very* lucky.

6.4 {183}

If the interferer's spectrum does not overlap that of our communications channel — the interferer is out-of-band — we need only use a bandpass filter that selects our transmission band and removes other portions of the spectrum.

6.5 {184}

The additive-noise channel is *not* linear because it does not have the zero-input-zero-output property (even though we might transmit nothing, the receiver's input consists of noise).

6.6 {188}

The signal-related portion of the transmitted spectrum is given by $X(f) = \frac{1}{2}S(f - f_c) + \frac{1}{2}S(f + f_c)$. Multiplying at the receiver by the carrier shifts this spectrum to $+f_c$ and $-f_c$, and scales the result by half.

$$\begin{aligned}\frac{1}{2}X(f - f_c) + \frac{1}{2}X(f + f_c) &= \left[\frac{1}{4}S(f - 2f_c) + \frac{1}{4}S(f)\right] + \left[\frac{1}{4}S(f) + \frac{1}{4}S(f + 2f_c)\right] \\ &= \frac{1}{4}S(f - 2f_c) + \frac{1}{2}S(f) + \frac{1}{4}S(f + 2f_c)\end{aligned}$$

The signal components centered at twice the carrier frequency are removed by the lowpass filter, while the baseband signal $S(f)$ emerges.

6.7 {188}

The key here is that the two spectra $S(f - f_c)$, $S(f + f_c)$ do not overlap because we have assumed that the carrier frequency f_c is much greater than the signal's highest frequency. Consequently, the term $S(f - f_c)S(f + f_c)$ normally obtained in computing the magnitude-squared equals zero.

6.8 {190}

Separation is $2W$. Commercial AM signal bandwidth is 5 kHz. Speech well contained in this bandwidth, much better than in the telephone!

6.9 {192}

$k = 4$.

6.10 {192}

$x(t) = \sum_n (-1)^{b(n)} A\, \mathrm{p}_T(t - nT) \sin\left(\frac{2\pi kt}{T}\right)$

6.11 {193}
The harmonic distortion is 10%.

6.12 {193}
Twice the baseband bandwidth because both positive and negative frequencies are shifted to the carrier by the modulation: $3R$.

6.13 {194}
We want to show that $f_1 - f_0 + 1/T > 3/2T$, which means that $f_1 - f_0 > 1/2T$ must be true. As we have $f_0 = k_0/T$, $f_1 = k_1/T$, the smallest difference between the two frequencies is $1/T$, giving a transmission bandwidth of $2/T$ or $2R$ in terms of data rate.

6.14 {196}
In BPSK, the signals are negatives of each other: $s_1(t) = -s_0(t)$. Consequently, the output of each multiplier-integrator combination is the negative of the other. Choosing largest therefore amounts to choosing which one is positive. We only need calculate one of these. If it's positive, we're done; if negative, we choose the other signal.

6.15 {196}
The matched-filter outputs are $\pm A^2T/2$ because the sinusoid has less power than a pulse having the same amplitude.

6.16 {199}
The noise-free integrator outputs differ by αA^2T, the factor of two smaller value than in the baseband case arising because the sinusoidal signals have less energy for the same amplitude. Stated in terms of E_b, the difference equals $2\alpha E_b$ just as in the baseband case.

6.17 {200}
The noise-free integrator output difference now equals $\alpha A^2T = \alpha E_b/2$. The noise power remains the same as in the BPSK case, which yields from Eq. 6.5 $p_e = Q\left(\sqrt{\alpha^2 E_b/\mathcal{N}_0}\right)$.

6.18 {202}

Equally likely symbols each have a probability of $1/K$. Thus, $\mathcal{H}[\mathcal{A}] = -\sum 1/K \log_2 1/K = \log_2 K$. To prove that this is the maximum-entropy probability assignment, we must explicitly take into account that probabilities sum to one. Focus on a particular symbol, say the first. $\Pr[a_0]$ appears *twice* in the entropy formula: the terms $\Pr[a_0] \log_2 \Pr[a_0]$ and $(1 - \Pr[a_0] - \Pr[a_1] - \cdots) \log_2(1 - \Pr[a_0] - \Pr[a_1] - \cdots)$. The derivative with respect to this probability (and all the others) must be zero. The derivative equals $\log \Pr[a_0] - \log(1 - \Pr[a_0] - \Pr[a_1] - \cdots)$, and all other derivatives have the same form (just substitute your letter's index). Thus, each probability must equal the others, and we are done.

For the minimum entropy answer, one term is $1 \log_2 1 = 0$, and the others are $0 \log_2 0$, which we define to be zero also. The minimum value of entropy is zero.

6.19 {206}

The Huffman coding tree for the second set of probabilities is *identical* to that for the first (Figure 6.12 {205}). The average code length is $\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{5} \cdot 3 + \frac{1}{20} \cdot 3 = 1.75$ bits. The entropy calculation is straightforward: $\mathcal{H}[=$ $] - \left(\frac{1}{2} \log \frac{1}{2} + \frac{1}{4} \log \frac{1}{4} + \frac{1}{5} \log \frac{1}{5} + \frac{1}{20} \log \frac{1}{20}\right)$, which equals 1.68 bits.

6.20 {206}

$T = 1/\overline{B}(\mathcal{A})R.$

6.21 {207}

Because no codeword begins with another's codeword, the first codeword encountered in a bit stream must be the right one. Note that we must start at the beginning of the bit stream; jumping into the middle does not guarantee perfect decoding. The end of one codeword and the beginning of another could be a codeword, and we would get lost.

6.22 {207}

Consider the bitstream $\ldots 0110111 \ldots$ taken from the bitstream $0|10|110|110|111|\ldots$. We would decode the initial part incorrectly, then would synchronize. If we had a fixed-length code (say 00, 01, 10, 11), the situation is *much* worse. Jumping into the middle leads to no synchronzation at all!

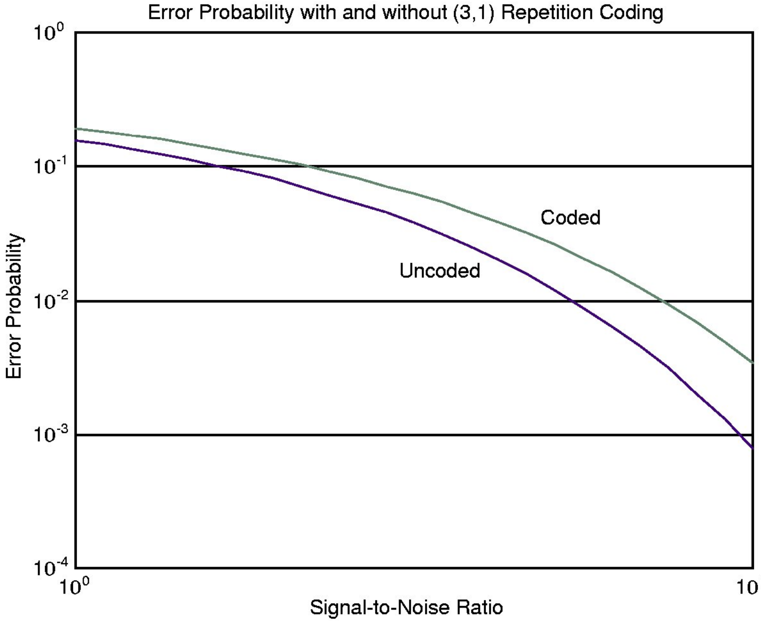
6.23 {209}

Is $3p_e^2(1 - p_e) + p_e^3 \overset{?}{\leq} p_e$? This question is equivalent to $3p_e(1 - p_e) + p_e^2 \overset{?}{\leq} 1$ or

$2p_e^2 - 3p_e + 1 \overset{?}{\geq} 0$. Because this is an upward-going parabola, we need only check where its roots are. Using the quadratic formula, we find that they are located at $\frac{1}{2}$ and 1. Consequently in the range $0 \leq p_e \leq \frac{1}{2}$ the error rate produced by coding is smaller.

6.24 {210}

With no coding, the average bit-error probability p_e is given by (6.7) {199}: $p_e = Q(\sqrt{2\alpha^2 E_b/\mathcal{N}_0})$. With a threefold repetition code, the bit-error probability is given by $3{p'_e}^2(1 - p'_e) + {p'_e}^3$, where $p'_e = Q(\sqrt{2\alpha^2 E_b/3\mathcal{N}_0})$. Plottng this reveals that the increase in bit-error probability out of the channel because of the energy reduction is not compensated by the repetition coding.



6.25 {212}

$d_{\text{min}} = 2n + 1$.

6.26 {213}

$\mathbf{G} = \text{col}[\mathtt{1}, \mathtt{1}, \mathtt{1}]$. As it has only one non-zero codeword and it has three ones, it has single-bit error correction capability. For double-bit error correction, we would need a five-fold repetition code.

6.27 {214}

When we multiply the parity-check matrix times any codeword equal to a column of $\mathbf{G}$, the result consists of the sum of an entry from the lower portion of $\mathbf{G}$ and itself that, by the laws of binary arithmetic, is always zero.

Because the code is linear—sum of any two codewords is a codeword—we can generate all codewords as sums of columns of $\mathbf{G}$. Since multiplying by $\mathbf{H}$ is also linear, $\mathbf{Hc} = \mathbf{0}$.

6.28 {215}

In binary arithmetic (Eq. (6.10) {210}), adding `0` to a binary value results in that binary value while adding `1` results in the opposite binary value.

6.29 {215}

The probability of a single-bit error in a length-N block is $Np_e(1-p_e)^{N-1}$ and a triple-bit error has probability $\binom{N}{3} p_e^3(1-p_e)^{N-3}$. For the first to be greater than the second, we must have $p_e < \sqrt{(N-1)(N-2)/6} + 1$. For $N = 7$, $p_e < 0.31$.

6.30 {216}

In a length-N block, N single-bit and $N(N-1)/2$ double-bit errors can occur. The number of non-zero vectors resulting from $\mathbf{H}\widehat{\mathbf{c}}$ must equal the sum of these two numbers.

$$2^{N-K} - 1 = N + N(N-1)/2 \quad \text{or} \quad 2^{N-K} = \frac{N^2 + N + 1}{2}$$

The first two solutions are $(5, 1)$ and $(90, 78)$ codes.

6.31 {219}

To convert to bits/second, we divide the capacity stated in bits/transmission by the bit interval duration T.

6.32 {221}

The network entry point is the telephone handset, which connects you to the nearest station. Dialing the telephone number informs the network of who will be the message recipient. The telephone system forms an electrical circuit between your handset and your friend's handset. Your friend receives the message via the same device—the handset—that served as the network entry point.

6.33 {225}
The transmitting op-amp sees a load or $R_{\text{out}} + Z_0 || (R_{\text{out}}/N)$, where N is the number of transceivers other than this one attached to the coaxial cable. The transfer function to some other transceiver's receiver circuit is R_{out} divided by this load.

6.34 {227}
The worst-case situation occurs when one computer begins to transmit just before the other's packet arrives. Transmitters must sense a collision before packet transmission ends. The time taken for one computer's packet to travel the Ethernet's length *and* for the other computer's transmission to arrive equals the round-trip, not one-way. propagation time.

6.35 {228}
The cable must be a factor of ten shorter: It cannot exceed 100 m. Different minimum packet sizes means different packet formats, making connecting old and new systems together more complex than need be.

6.36 {228}
When you pick up the telephone, you initiate a dialog with your network interface by dialing the number. The network looks up where the destination corresponding to that number is located, and routes the call accordingly. The route remains fixed as long as the call persists. What you say amounts to high-level protocol while establishing the connection and maintaining it corresponds to low-level protocol.

A.1 {240}
To convert $3 - 2j$ to polar form, we first locate the number in the complex plane in the fourth quadrant. The distance from the origin to the complex number is the magnitude r, which equals $\sqrt{13} = \sqrt{3^2 + (-2)^2}$. The angle equals $-\tan^{-1}\frac{2}{3}$ or -0.588 radians ($-33.7°$). The final answer is $\sqrt{13}\angle{-33.7°}$.

A.2 {241}
$z + z^* = a + jb + (a - jb) = 2a = 2\text{Re}[z]$. Similarly, $z - z^* = a + jb - (a - jb) = 2jb = 2j\text{Im}[z]$.

A.3 {241}
$zz^* = (a + jb)(a - jb) = a^2 + b^2$. Thus, $zz^* = r^2$.

A.4 {241}

$$\frac{1+j}{1-j}=\frac{(1+j)(1+j)}{2}=\frac{2j}{2}=j$$

B.1 {243}

Alexander Graham Bell. He developed it because we seem to perceive physical quantities like loudness and brightness logarithmically. In other words, *percentage*, not absolute differences, matter to us. We use decibels today because common values are small integers. If we used Bels, they would be decimal fractions, which aren't as elegant.